

```

EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDDDD FFFFFFFFFFFFFF
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDDDD FFFFFFFFFFFFFF
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDDDD FFFFFFFFFFFFFF
EEE DDD DDD FFF
EEE DDD DDD FFF
EEE DDD DDD FFF
EEE DDD DDD FFF
EEE DDD DDD FFF
EEE DDD DDD FFF
EEE DDD DDD FFF
EEEEEEEEEEEEEEEEEE DDD DDD FFF
EEEEEEEEEEEEEEEEEE DDD DDD FFFFFFFF
EEEEEEEEEEEEEEEEEE DDD DDD FFFFFFFF
EEEEEEEEEEEEEEEEEE DDD DDD FFFFFFFF
EEE DDD DDD FFF
EEE DDD DDD FFF
EEE DDD DDD FFF
EEE DDD DDD FFF
EEE DDD DDD FFF
EEE DDD DDD FFF
EEEEEEEEEEEEEEEEEE DDD DDD FFF
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDD FFF
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDD FFF
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDD FFF

```

```

LL          IIIIII      SSSSSSSS
LL          IIIIII      SSSSSSSS
LL          II         SS
LL          II         SS
LL          II         SS
LL          II         SS
LL          II         SSSSSS
LL          II         SSSSSS
LL          II         SS
LL          II         SS
LL          II         SS
LL          II         SS
LLLLLLLLLLL IIIIIIII   SSSSSSSS
LLLLLLLLLLL IIIIIIII   SSSSSSSS

```

EDF  
V04-000

Source Listing

E 10  
16-Sep-1984 01:22:54  
5-Sep-1984 13:37:22

VAX-11 Pascal V2.4-277  
DISK\$VMSMASTER:[EDF.SRC]EDFMAIN.PAS;1 (1)

Page 1

0001  
0002  
0003  
0004  
0005  
0006  
0007  
0008  
0009  
0010  
0011  
0012  
0013  
0014  
0015  
0016  
0017  
0018  
0019  
0020  
0021  
0022  
0023  
0024  
0025  
0026  
0027  
0028  
0029  
0030  
0031  
0032  
0033  
0034  
0035  
0036  
0037  
0038  
0039  
0040  
0041  
0042  
0043  
0044  
0045  
0046  
0047  
0048  
0049  
0050  
0051  
0052  
0053  
0054  
0055  
0056  
0057

```
[ IDENT ('V04-000'),  
( ++  
*****  
**  
**  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY  
**  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.  
**  ALL RIGHTS RESERVED.  
**  
**  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
**  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
**  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
**  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
**  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
**  TRANSFERRED.  
**  
**  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
**  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
**  CORPORATION.  
**  
**  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
**  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.  
**  
*****
```

FACILITY: VAX/VMS EDF (EDIT/FDL) UTILITY

ABSTRACT: This facility is used to create, modify, and optimize  
FDL specification files.

ENVIRONMENT: NATIVE/USER MODE

AUTHOR: Ken F. Henderson Jr.

CREATION DATE: 27-Mar-1981

MODIFIED BY:

V03-011	KFH0011	Ken Henderson	8 Aug 1983
	Changes for separate compilation.		
V03-010	KFH0010	Ken Henderson	26 Apr 1983
	Added ADD_KEY, DELETE_KEY scripts. Transferred some initializations from INIT_EDITOR to EDFVAR. Changed 'redesign' to 'touchup'.		
V03-009	KFH0009	Ken Henderson	14 Apr 1983
	Added SET_FUNCTION, RESPONSES, GRANULARITY, PROMPTING, JOURNAL_ENABLED.		
V03-008	KFH0008	Ken Henderson	20 Jan 1983
	Removed references to DASH.		



### Source Listing

F 10  
16-Sep-1984 01:22:54  
5-Sep-1984 13:37:22

VAX-11 Pascal V2.4-277 Page  
DISK\$VMSMASTER:[EDF.SRC]EDFMAIN.PAS:1 (1)

Page 2

0058  
0059  
0060  
0061  
0062  
0063  
0064  
0065  
0066  
0067  
0068  
0069  
0070  
0071  
0072  
0073  
0074  
0075  
0076  
0077  
0078  
0079  
0080  
0081  
0082  
0083  
0084  
0085

-- }

V03-007	KFH0007	Ken Henderson	30 Dec 1982
	Finished support of Pascal V2.		
V03-006	KFH0006	Ken Henderson	22 Nov 1982
	Added support for Pascal V2. Added support for additional FILE and CONNECT attributes.		
V03-005	KFH0005	Ken Henderson	8 Sept 1982
	Modified references to many variables to fit with database reorganization.		
V03-004	KFH0003	Ken Henderson	26-Mar-1982
	Modified Mainline code to establish the CTRLZ handler before asking to continue after an error parsing the FDL definition file - QAR 885.		
V03-002	KFH0002	Ken Henderson	23-Mar-1982
	Modified routines INPUT_FDL_FILE and INPUT_ANALYSIS_FILE and the mainline code to fix FT2 QARs 694,699		
V03-001	KFH0001	Ken Henderson	17-Mar-1982
	Reset TEMP_FULL_PROMPT to false in the main loop. (dispatch_function)		

[illegible]

EDF  
V04-000

Source Listing

G 10  
16-Sep-1984 01:22:54  
5-Sep-1984 13:37:22

VAX-11 Pascal V2.4-277  
DISK\$VMSMASTER:[EDF.SRC]EDFMAIN.PAS;1 (2) Page 3

```
0087      ( ++
0088      This is the list of environments
0089      -- )
0090
0091      INHERIT (
0092
0093      'SYSSLIBRARY:STARLET',      { System definitions }
0094      'SHRLIBS:FDLPARDEF',        { FDL facility definitions }
0095      'SHRLIBS:FDLSDLMSG',        { FDL Message definitions }
0096      'LIBS:EDFSTRUCT',          { EDF Tparse and other definitions }
0097      'LIBS:EDFSDLMSG',          { EDF Message definitions }
0098
0099      'LIBS:EDFCONST',
0100      'LIBS:EDFTYPE',
0101      'LIBS:EDFVAR',
0102      'LIBS:EDFEXTERN',
0103      'LIBS:EDFCHF',
0104      'LIBS:EDFUTIL',
0105      'LIBS:EDFASK',
0106      'LIBS:EDFSHOW',
0107      'LIBS:EDFDESIGN',
0108      'LIBS:EDFFUNCS'
0109
0110      )]
0111
0112      PROGRAM EDF (INPUT,OUTPUT);
```

EDF  
V04-000

Source Listing

H 10  
16-Sep-1984 01:22:54  
5-Sep-1984 13:37:22

VAX-11 Pascal V2.4-277  
DISK\$VMSMASTER:[EDF.SRC]EDFMAIN.PAS;1 (3)

Page 4

```
0114 { ++
0115
0116 INIT_EDITOR -- Initialize EDF upon startup.
0117
0118 This is the setup code to create the initial conditions for EDF.
0119
0120 CALLING SEQUENCE:
0121
0122 INIT_EDITOR;
0123
0124 INPUT PARAMETERS:
0125
0126 none
0127
0128 IMPLICIT INPUTS:
0129
0130 DCL (through the CLIS routines)
0131
0132 OUTPUT PARAMETERS:
0133
0134 none
0135
0136 IMPLICIT OUTPUTS:
0137
0138 EDITING
0139 ANSI_RESET
0140 ANSI_BOLD
0141 ANSI_UNDERSCORE
0142 ANSI_BLINK
0143 ANSI_REVERSE
0144 NULL_STRING4
0145 SHIFT
0146 LOW_SHIFT
0147 TERMINAL_SPEED
0148 RMS_INPUT_ERROR
0149 VID_TERM
0150 DEV_TYPE
0151 LINE_WIDTH
0152 LINES_PER_PAGE
0153 VIDEO_TERMINAL
0154 NULL_CHAR
0155 CONTROL_G
0156 CONTROL_W
0157 CONTROL_Z
0158 TAB
0159 CRLF
0160 ESCAPE
0161 QUESTION_MARK
0162 FDL_BLOCK
0163 FDL$AL_BLOCK
0164 EDF$GL_PROT_MASK
0165 EDF$GL_FID1
0166 EDF$GL_FID2
0167 EDF$GL_FID3
0168 EDF$GL_UIC_GROUP
0169 EDF$GL_UIC_MEMBER
0170 EDF$AB_STRING
```

EDF  
V04-000

Source Listing

I 10  
16-Sep-1984 01:22:54  
5-Sep-1984 13:37:22

VAX-11 Pascal V2.4-277  
DISK\$VMSMASTER:[EDF.SRC]EDFMAIN.PAS;1 (3)

Page 5

```
0171 EDF$AB COMMENT
0172 LIB$PUT_OUTPUT_PTR
0173 LIB$GET_INPUT_PTR
0174 DEF_CURRENT
0175 DEF_HEAD
0176 DEF_TAIL
0177 DEF_PRED
0178 DEF_SUCC
0179 INPUT_FILENAME_DESC
0180 OUTPUT_FILENAME_DESC
0181 ANALYSIS_FILENAME_DESC
0182 IDATA[EDF$K_FIRST_SCRIPT]
0183 FULL_PROMPT
0184
0185 ROUTINES CALLED:
0186
0187 EDF$TERM_SETUP
0188 ESTABLISH
0189 LIB$SIGNAL
0190 CLISGET_VALUE
0191 CLISPRESENT
0192
0193 ROUTINE VALUE:
0194
0195 none
0196
0197 SIGNALS:
0198
0199 EDF$_SMALLPAGE - if term screen size too small
0200
0201 SIDE EFFECTS:
0202
0203 EDF is initialized.
0204
0205 -- }
```



```
0207 PROCEDURE INIT_EDITOR;
```

```
0208 VAR
0209     TEMP_DESCRIPTOR      : DESCRIPTOR;
0210     I                    : INTEGER;
```

```
0211 BEGIN
```

```
0212 { +
0213 See if the user wants batch mode or interactive.
0214 - }
0215 IF NOT (ODD (CLIPRESENT ('INTERACTIVE')) THEN
```

```
0216 { +
0217 At this point, the user has specified /NOINTERACTIVE and
0218 wants a quick, automatic tuneup for his file.
```

```
0219 - }
0220 BEGIN
```

```
0221 TAKE_DEFAULTS      := TRUE;
0222 AUTO_TUNE          := TRUE;
0223 JOURNAL_ENABLED    := FALSE;
0224 QTAB[EDFSK_RETURN].DEFAULT_OK := TRUE;
0225 IDATA[EDFSK_RESPONSES] := EDFSK_AUTO;
0226 IDATA[EDFSK_FIRST_SCRIPT] := EDFSK_OPTIMIZE_FDL;
0227 QTAB[EDFSK_CURRENT_FUNCTION].DEFAULT := EDFSK_QUIT;
0228 QTAB[EDFSK_DESIGN_CYCLE].DEFAULT := EDFSK_FINIS;
0229 VIDEO_TERMINAL     := FALSE;
0230 DEC_CRT            := FALSE;
0231 ANSI_CRT           := FALSE;
0232 REGIS             := FALSE;
```

```
0233 END      ( IF TRUE NOT (ODD (CLIPRESENT ('INTERACTIVE')) )
```

```
0234 ELSE
```

```
0235 BEGIN
```

```
0236 { +
0237 Do initialization on the terminal. Get its speed, setup exit handler,
0238 Also check to make sure that indeed the
0239 input is a terminal (and STOPS if not) and if the terminal isn't a scope,
0240 then it sets the page length to 16
0241 (as required by hardcopy surface plots).
```

```
0242 *****
0243 The call to EDF$TERM_SETUP Must come BEFORE ANY calls to the
0244 SCREEN PACKAGE!!!
0245 *****
```

```
0246 - }
0247 TERMINAL_SPEED := EDF$TERM_SETUP;
```

```
0248 { +
0249 EDF$TERM_SETUP returns a status of EDF$DEVCLASS if SYSSINPUT is
0250 not a terminal. In that case, see if the magic logical name
0251 EDF$PLAYBACK_INPUT is defined. If so, then set the terminal speed
```



```
0264 to be 2400 baud and continue, if the logical is not defined,
0265 exit with the DEVCLASS status.
0266 NOTE THAT THE LOGICAL NAME 'EDF$$PLAYBACK_INPUT' IS NOT SUPPORTED
0267 FOR CUSTOMERS AND IS USED INTERNALLY ONLY FOR REGRESSION TESTING!!!
0268 - )
0269 IF LIB$MATCH_COND (TERMINAL_SPEED,EDF$_DEVCLASS) THEN
0270 BEGIN
0271     TEMP_STATUS := $TRNLOG ('EDF$$PLAYBACK_INPUT',TEMP_STRING255);
0272     IF LIB$MATCH_COND (TEMP_STATUS,SS$_NOTRAN) THEN
0273         LIB$STOP (EDF$_DEVCLASS,0,0,0)
0274     ELSE IF LIB$MATCH_COND (TEMP_STATUS,SS$_NORMAL) THEN
0275         TERMINAL_SPEED := TT$C_BAUD_2400;
0276     END;
0277 { +
0278 Set up ^C ast routine.
0279 - )
0280 EDF$CTRLCAST;
0281 { +
0282 First find out what terminal characteristics we have.
0283 - )
0284 LIB$SCREEN_INFO (
0285     SCREEN_FLAGS,
0286     DEV_TYPE,
0287     LINE_WIDTH,
0288     LINES_PER_PAGE
0289 );
0290 { +
0291 Instead of using the actual terminal type,
0292 we'll just test on decrt or not.
0293 - )
0294 VIDEO_TERMINAL := SCREEN_FLAGS.SCR$V_SCREEN;
0295 DEC CRT := SCREEN_FLAGS.SCR$V_DECCRT;
0296 ANSI CRT := SCREEN_FLAGS.SCR$V_ANSICRT;
0297 REGIS := SCREEN_FLAGS.SCR$V_REGIS;
0298 { +
0299 See if we have enough room on the terminal.
0300 - )
0301 IF (
0302     (LINE_WIDTH < MINIMUM_TERM_WIDTH)
0303 OR
0304     (VIDEO_TERMINAL AND (LINES_PER_PAGE < MINIMUM_VIDEO_PAGE))
0305 ) THEN
0306     { +
0307 Not enough room!
0308 - )
```

```
0321      LIB$SIGNAL (EDF$_SMALLPAGE,2,LIN_WIDTH,LIN_PER_PAGE);
0322
0323      OPEN      (OUTPUT,SYS$OUTPUT_NAME,NEW,RECORD_LENGTH := 2148);
0324      REWRITE (OUTPUT);
0325
0326      { +
0327      Open the journal file, if it's requested.
0328      NOTE THAT THE LOGICAL NAME 'EDF$$JOURNAL_INPUT' IS NOT SUPPORTED FOR
0329      CUSTOMERS, AND IS USED INTERNALLY ONLY FOR REGRESSION TESTING.
0330      - }
0331      TEMP_STATUS      := $TRNLOG ('EDF$$JOURNAL_INPUT',,JOURNAL_FILENAME);
0332
0333      JOURNAL_ENABLED := (
0334                          (LIB$MATCH_COND (TEMP_STATUS,SS$_NORMAL))
0335                          AND
0336                          (NOT LIB$MATCH_COND (TEMP_STATUS,SS$_NOTRAN))
0337                          );
0338
0339      IF JOURNAL_ENABLED THEN
0340      BEGIN
0341
0342          OPEN      (
0343              FILE_VARIABLE := JOURNAL_FILE,
0344              FILE_NAME := JOURNAL_FILENAME,
0345              HISTORY := NEW,
0346              RECORD_LENGTH := 255,
0347              RECORD_TYPE := VARIABLE
0348          );
0349          REWRITE      (JOURNAL_FILE);
0350
0351      END;
0352
0353      END;      { IF FALSE (NOT ODD (CLIS$PRESENT ('INTERACTIVE')) ) }
0354
0355      { +
0356      If we don't have an ANSI terminal (VT100-series and up), then zero out the
0357      video attribute arrays.
0358      - }
0359      IF NOT DEC_CRT THEN
0360      BEGIN
0361
0362          FOR I := 1 TO 4 DO
0363          BEGIN
0364              ANSI_RESET[I]      := NULL_CHAR;
0365              ANSI_BOLD[I]       := NULL_CHAR;
0366              ANSI_UNDERSCORE[I] := NULL_CHAR;
0367              ANSI_BLINK[I]      := NULL_CHAR;
0368              ANSI_REVERSE[I]    := NULL_CHAR;
0369
0370          END;
0371
0372      END;
0373
0374      END;      { IF NOT DEC_CRT }
```

```
0378      { +
0379      If we have more than 80 chars per line, we may have to shift everything
0380      over to the right. (if we also have a video terminal)
0381      - }
0382      IF (LINE_WIDTH > EDFSC_SHIFTPOINT) AND VIDEO_TERMINAL THEN
0383
0384      BEGIN
0385
0386          SHIFT[2]           := TAB;
0387          SHIFT[3]           := TAB;
0388          SHIFT[4]           := TAB;
0389          CRLF_SHIFT[4]      := TAB;
0390          CRLF_SHIFT[5]      := TAB;
0391          CRLF_SHIFT[6]      := TAB;
0392
0393      END;
0394
0395      { +
0396      The 'under-graph' text comes out in graphics mode for Regis devices.
0397      - }
0398      IF REGIS THEN
0399
0400      BEGIN
0401
0402          LOW_SHIFT[1]       := NULL_CHAR;
0403          LOW_SHIFT[2]       := NULL_CHAR;
0404          LOW_SHIFT[3]       := NULL_CHAR;
0405
0406      END      ( IF TRUE REGIS )
0407
0408      ELSE
0409
0410      BEGIN
0411
0412          LOW_SHIFT[1]       := SHIFT[2];
0413          LOW_SHIFT[2]       := SHIFT[3];
0414          LOW_SHIFT[3]       := SHIFT[4];
0415
0416      END;      ( IF FALSE REGIS )
0417
0418      { +
0419      OK, so let the user know that we're here.
0420      - }
0421      CLEAR (SCREEN);
0422
0423      { +
0424      Initialize the TPARSE block.
0425      - }
0426      WITH PARAM_BLOCK DO
0427
0428      BEGIN
0429
0430          TPASL_COUNT      := TPASK_COUNT0;
0431          TPASV_ABBREV      := TRUE;
0432
0433      END;      ( DO )
```



```
0435 { +
0436 Stuff the pointer variable FDL_BLOCK with the address of FDL$AL_BLOCK
0437 PLUS the Contents of FDL$AL_BLOCK. The offset is introduced by the
0438 transfer vector in the shareable image FDL$HR.EXE. (home of FDL$AL_BLOCK)
0439 - }
0440 FDL_BLOCK::INTEGER := IADDRESS (FDL$AL_BLOCK) + FDL$AL_BLOCK;
0441
0442 { +
0443 Now stuff the address of our EDF$LINE_PARSED routine into the callback
0444 address cell in the FDL$AL_BLOCK array.
0445 - }
0446 FDL_BLOCK^[FDL$AL_PCALL] := IADDRESS (EDF$LINE_PARSED);
0447
0448 { +
0449 Setup some defaults.
0450 - }
0451 IDATA[EDF$K_RESPONSES] := EDF$K_MAN;
0452 BDATA[EDF$K_BLOCK_SPAN] := TRUE;
0453 IDATA[EDF$K_BUCKET_WEIGHT] := EDF$K_FLATTER_FILES;
0454
0455 { +
0456 This initializes the the QTAB table with the addresses of the TParse tables.
0457 - }
0458 QTAB[EDF$K_CARR_CTRL].KEY_TABLE := IADDRESS (EDF$AB_CARR_TABLE_KEY);
0459 QTAB[EDF$K_CARR_CTRL].STATE_TABLE := IADDRESS (EDF$AB_CARR_TABLE_STA);
0460 QTAB[EDF$K_RECORD_FORMAT].KEY_TABLE := IADDRESS (EDF$AB_FORMAT_TABLE_KEY);
0461 QTAB[EDF$K_RECORD_FORMAT].STATE_TABLE := IADDRESS (EDF$AB_FORMAT_TABLE_STA);
0462 QTAB[EDF$K_KEY_TYPE].KEY_TABLE := IADDRESS (EDF$AB_TYPE_TABLE_KEY);
0463 QTAB[EDF$K_KEY_TYPE].STATE_TABLE := IADDRESS (EDF$AB_TYPE_TABLE_STA);
0464 QTAB[EDF$K_LOAD_METHOD].KEY_TABLE := IADDRESS (EDF$AB_LOAD_METHOD_TABLE_KEY);
0465 QTAB[EDF$K_LOAD_METHOD].STATE_TABLE := IADDRESS (EDF$AB_LOAD_METHOD_TABLE_STA);
0466 QTAB[EDF$K_BUCKET_WEIGHT].KEY_TABLE := IADDRESS (EDF$AB_WEIGHT_TABLE_KEY);
0467 QTAB[EDF$K_BUCKET_WEIGHT].STATE_TABLE := IADDRESS (EDF$AB_WEIGHT_TABLE_STA);
0468 QTAB[EDF$K_SURFACE_OPTION].KEY_TABLE := IADDRESS (EDF$AB_SURFACE_OPTION_TABLE_KEY);
0469 QTAB[EDF$K_SURFACE_OPTION].STATE_TABLE := IADDRESS (EDF$AB_SURFACE_OPTION_TABLE_STA);
0470 QTAB[EDF$K_CURRENT_FUNCTION].KEY_TABLE := IADDRESS (EDF$AB_CURRENT_FUNC_TABLE_KEY);
0471 QTAB[EDF$K_CURRENT_FUNCTION].STATE_TABLE := IADDRESS (EDF$AB_CURRENT_FUNC_TABLE_STA);
0472 QTAB[EDF$K_DESIGN_CYCLE].KEY_TABLE := IADDRESS (EDF$AB_DESIGN_CYCLE_TABLE_KEY);
0473 QTAB[EDF$K_DESIGN_CYCLE].STATE_TABLE := IADDRESS (EDF$AB_DESIGN_CYCLE_TABLE_STA);
0474 QTAB[EDF$K_SCRIPT_OPTION].KEY_TABLE := IADDRESS (EDF$AB_SCRIPT_OPTION_TABLE_KEY);
0475 QTAB[EDF$K_SCRIPT_OPTION].STATE_TABLE := IADDRESS (EDF$AB_SCRIPT_OPTION_TABLE_STA);
0476 QTAB[EDF$K_KEY_COMP_WANTED].KEY_TABLE := IADDRESS (EDF$AB_YES_NO_TABLE_KEY);
0477 QTAB[EDF$K_KEY_COMP_WANTED].STATE_TABLE := IADDRESS (EDF$AB_YES_NO_TABLE_STA);
0478 QTAB[EDF$K_REC_COMP_WANTED].KEY_TABLE := IADDRESS (EDF$AB_YES_NO_TABLE_KEY);
0479 QTAB[EDF$K_REC_COMP_WANTED].STATE_TABLE := IADDRESS (EDF$AB_YES_NO_TABLE_STA);
0480 QTAB[EDF$K_IDX_COMP_WANTED].KEY_TABLE := IADDRESS (EDF$AB_YES_NO_TABLE_KEY);
0481 QTAB[EDF$K_IDX_COMP_WANTED].STATE_TABLE := IADDRESS (EDF$AB_YES_NO_TABLE_STA);
0482 QTAB[EDF$K_CONFIRM].KEY_TABLE := IADDRESS (EDF$AB_YES_NO_TABLE_KEY);
0483 QTAB[EDF$K_CONFIRM].STATE_TABLE := IADDRESS (EDF$AB_YES_NO_TABLE_STA);
0484 QTAB[EDF$K_BLOCK_SPAN].KEY_TABLE := IADDRESS (EDF$AB_YES_NO_TABLE_KEY);
0485 QTAB[EDF$K_BLOCK_SPAN].STATE_TABLE := IADDRESS (EDF$AB_YES_NO_TABLE_STA);
0486 QTAB[EDF$K_ASCENDING_ADDED].KEY_TABLE := IADDRESS (EDF$AB_YES_NO_TABLE_KEY);
0487 QTAB[EDF$K_ASCENDING_ADDED].STATE_TABLE := IADDRESS (EDF$AB_YES_NO_TABLE_STA);
0488 QTAB[EDF$K_ASCENDING_LOAD].KEY_TABLE := IADDRESS (EDF$AB_YES_NO_TABLE_KEY);
0489 QTAB[EDF$K_ASCENDING_LOAD].STATE_TABLE := IADDRESS (EDF$AB_YES_NO_TABLE_STA);
0490 QTAB[EDF$K_RETURN].KEY_TABLE := IADDRESS (EDF$AB_YES_NO_TABLE_KEY);
0491 QTAB[EDF$K_RETURN].STATE_TABLE := IADDRESS (EDF$AB_YES_NO_TABLE_STA);
```

EDF  
V04-000

Source Listing

B 11  
16-Sep-1984 01:22:54  
5-Sep-1984 13:37:22

VAX-11 Pascal V2.4-277  
DISK\$VMSMASTER:[EDF.SRC]EDFMAIN.PAS;1 (5) Page 11

```
0492 QTAB[EDFSK_KEY_DUPS].KEY TABLE      := IADDRESS (EDFSAB_YES_NO_TABLE_KEY);
0493 QTAB[EDFSK_KEY_DUPS].STATE TABLE     := IADDRESS (EDFSAB_YES_NO_TABLE_STA);
0494 QTAB[EDFSK_KEY_DIST].KEY TABLE       := IADDRESS (EDFSAB_YES_NO_TABLE_KEY);
0495 QTAB[EDFSK_KEY_DIST].STATE TABLE     := IADDRESS (EDFSAB_YES_NO_TABLE_STA);
0496 QTAB[EDFSK_KEY_CHANGES].KEY TABLE   := IADDRESS (EDFSAB_YES_NO_TABLE_KEY);
0497 QTAB[EDFSK_KEY_CHANGES].STATE TABLE := IADDRESS (EDFSAB_YES_NO_TABLE_STA);
0498 QTAB[EDFSK_SEGMENTED].KEY TABLE      := IADDRESS (EDFSAB_YES_NO_TABLE_KEY);
0499 QTAB[EDFSK_SEGMENTED].STATE TABLE    := IADDRESS (EDFSAB_YES_NO_TABLE_STA);
0500 QTAB[EDFSK_GLOBAL_WANTED].KEY TABLE  := IADDRESS (EDFSAB_YES_NO_TABLE_KEY);
0501 QTAB[EDFSK_GLOBAL_WANTED].STATE TABLE := IADDRESS (EDFSAB_YES_NO_TABLE_STA);
0502 QTAB[EDFSK_TEST_PRIMARY].KEY TABLE   := IADDRESS (EDFSAB_PRIMARY_TABLE_KEY);
0503 QTAB[EDFSK_TEST_PRIMARY].STATE TABLE := IADDRESS (EDFSAB_PRIMARY_TABLE_STA);
0504 QTAB[EDFSK_SET_FUNCTION].KEY TABLE   := IADDRESS (EDFSAB_SET_FUNCTION_TABLE_KEY);
0505 QTAB[EDFSK_SET_FUNCTION].STATE TABLE := IADDRESS (EDFSAB_SET_FUNCTION_TABLE_STA);
0506 QTAB[EDFSK_GRANULARITY].KEY TABLE    := IADDRESS (EDFSAB_GRANULARITY_TABLE_KEY);
0507 QTAB[EDFSK_GRANULARITY].STATE TABLE  := IADDRESS (EDFSAB_GRANULARITY_TABLE_STA);
0508 QTAB[EDFSK_PROMPTING].KEY TABLE      := IADDRESS (EDFSAB_PROMPTING_TABLE_KEY);
0509 QTAB[EDFSK_PROMPTING].STATE TABLE    := IADDRESS (EDFSAB_PROMPTING_TABLE_STA);
0510 QTAB[EDFSK_RESPONSES].KEY TABLE      := IADDRESS (EDFSAB_RESPONSES_TABLE_KEY);
0511 QTAB[EDFSK_RESPONSES].STATE TABLE    := IADDRESS (EDFSAB_RESPONSES_TABLE_STA);
```

ED  
V0

10  
10  
11  
11  
11  
11  
11  
11  
11  
11  
11  
11  
11

```
0513 { +
0514 Create an Ident line_object and put it into the list.
0515 - }
0516 NEW_IDENT_LINE;
0517
0518 { +
0519 See what we have.
0520 - }
0521 ANALYSIS_SPECIFIED := ODD (CLISPRESENT ('ANALYSIS'));
0522
0523 IF ANALYSIS_SPECIFIED THEN
0524 BEGIN
0525     { +
0526     Save the analysis filename.
0527     - }
0528     ANALYSIS_FILENAME_DESC := NULL_STRING;
0529     CLISGET_VALUE ('ANALYSIS', ANALYSIS_FILENAME_DESC);
0530
0531 END;      { IF ANALYSIS_SPECIFIED }
0532
0533 { +
0534 Save the input filename.
0535 - }
0536 INPUT_FILENAME_DESC := NULL_STRING;
0537 CLISGET_VALUE ('P1', INPUT_FILENAME_DESC);
0538
0539 { +
0540 Find out which output filename we're using, the /OUTPUT, or the
0541 command parameter.
0542 - }
0543 OUTPUT_FILENAME_DESC := NULL_STRING;
0544
0545 IF ODD (CLISPRESENT ('OUTPUT')) THEN
0546 BEGIN
0547     { +
0548     The /OUTPUT switch overrides, so use it if present.
0549     - }
0550     CLISGET_VALUE ('OUTPUT', OUTPUT_FILENAME_DESC);
0551
0552 END
0553 ELSE
0554     { +
0555     The user just wants another version of the input file.
0556     - }
0557     LIB$COPY_DXDX (INPUT_FILENAME_DESC, OUTPUT_FILENAME_DESC);
0558
0559 { +
0560 The following qualifiers make sense only if we're in normal
0561 interactive mode.
0562 - }
0563 IF ODD (CLISPRESENT ('INTERACTIVE')) THEN
```



```
0570
0571 BEGIN
0572
0573 { +
0574 Set up the script to the one specified in the DCL command. (if any)
0575 - }
0576 IF ODD (CLISPRESENT('SCRIPT')) THEN
0577 BEGIN
0578     TEMP_DESCRIPTOR := NULL_STRING;
0579     CLISGET_VALUE ('SCRIPT',TEMP_DESCRIPTOR);
0580
0581     { +
0582     Case on the 1st letter.
0583     - }
0584     CASE TEMP_DESCRIPTOR.DSCSA_POINTER^[1] OF
0585         'A' : IDATA[EDFSK_FIRST_SCRIPT] := EDFSK_ADD_KEY_FDL;
0586         'D' : IDATA[EDFSK_FIRST_SCRIPT] := EDFSK_DELETE_KEY_FDL;
0587         'I' : IDATA[EDFSK_FIRST_SCRIPT] := EDFSK_IDX_DESIGN_FDL;
0588         'S' : IDATA[EDFSK_FIRST_SCRIPT] := EDFSK_SEQ_DESIGN_FDL;
0589         'O' : IDATA[EDFSK_FIRST_SCRIPT] := EDFSK_OPTIMIZE_FDL;
0590         'R' : IDATA[EDFSK_FIRST_SCRIPT] := EDFSK_REL_DESIGN_FDL;
0591         'T' : IDATA[EDFSK_FIRST_SCRIPT] := EDFSK_REDESIGN_FDL;
0592
0593     OTHERWISE
0594         { +
0595         If the user blows it, give him nothing.
0596         - }
0597         IDATA[EDFSK_FIRST_SCRIPT] := EDFSK_ZERO_SCRIPT;
0598
0599     END;      { CASE }
0600     STR$FREE1_DX (TEMP_DESCRIPTOR);
0601
0602     END      { IF TRUE SCRIPT PRESENT }
0603 ELSE
0604     IDATA[EDFSK_FIRST_SCRIPT] := EDFSK_ZERO_SCRIPT;
0605
0606     { +
0607     Find out how many keys the user wants.
0608     - }
0609     IF ODD (CLISPRESENT('NUMBER_KEYS')) THEN
0610     BEGIN
0611         TEMP_DESCRIPTOR := NULL_STRING;
```

```
0627 CLISGET_VALUE ('SCRIPT',TEMP_DESCRIPTOR);
0628 ISTATUS := OTSSCVT Till (TEMP_DESCRIPTOR,
0629 IDATA[EDFSK_NUMBER_KEYS]);
0630 QTAB[EDFSK_NUMBER_KEYS].DEFAULT := IDATA[EDFSK_NUMBER_KEYS];
0631 NUMBER_KEYS_SET := TRUE;
0632
0633 STR$FREE1_DX (TEMP_DESCRIPTOR);
0634
0635 END;          ( IF TRUE SCRIPT PRESENT )
0636
0637 { +
0638 If the user specified a prompt level, set EDF's level to that,
0639 otherwise set it according to the type of terminal (hardcopy gets Brief).
0640 - }
0641 IF ODD (CLISPRESENT ('PROMPTING')) THEN
0642 BEGIN
0643   { +
0644   Get the prompting level specified by the user.
0645   - }
0646   TEMP_DESCRIPTOR := NULL STRING;
0647   CLISGET_VALUE ('PROMPTING',TEMP_DESCRIPTOR);
0648
0649   { +
0650   The 1st character of the string is unique.
0651   - }
0652   CASE TEMP_DESCRIPTOR.DSC$A_POINTER^ [1] OF
0653   ( +
0654   Brief prompting
0655   - )
0656   'B' :
0657     FULL_PROMPT := FALSE;
0658
0659   ( +
0660   Full prompting
0661   - )
0662   'F' :
0663     FULL_PROMPT := TRUE;
0664
0665   OTHERWISE
0666     { +
0667     Automatic prompting.
0668     Default to Brief prompting for non-scope (or slow) terminals.
0669     - }
0670     IF ( VIDEO_TERMINAL
0671         AND
0672         ( TERMINAL_SPEED >= TTSC_BAUD_2400 ) ) THEN
0673       FULL_PROMPT := TRUE
0674     ELSE
0675       FULL_PROMPT := FALSE;
0676
0677   ELSE
0678     FULL_PROMPT := FALSE;
0679
0680   ELSE
0681     FULL_PROMPT := FALSE;
0682
0683   ELSE
0684     FULL_PROMPT := FALSE;
```

EDF  
V04-000

Source Listing

F 11  
16-Sep-1984 01:22:54  
5-Sep-1984 13:37:22

VAX-11 Pascal V2.4-277  
DISK\$VMSMASTER:[EDF.SRC]EDFMAIN.PAS;1 (6) Page 15

```
0684         FULL_PROMPT := FALSE;
0685
0686     END;          ( CASE )
0687
0688     STR$FREE1_DX (TEMP_DESCRIPTOR);
0689
0690 END          ( IF TRUE (ODD) )
0691
0692 ELSE
0693
0694 BEGIN
0695     ( *
0696     Default to Brief prompting for non-scope (or slow) terminals.
0697     - )
0698     IF VIDEO_TERMINAL AND (TERMINAL_SPEED >= TT$C_BAUD_2400) THEN
0699         FULL_PROMPT      := TRUE
0700     ELSE
0701         FULL_PROMPT      := FALSE;
0702
0703     END;    ( IF FALSE (ODD) )
0704
0705     IF ODD (CL$PRESENT ('DISPLAY')) THEN
0706     BEGIN
0707         CL$GET_VALUE ('DISPLAY',TEMP_DESCRIPTOR);
0708
0709         CASE TEMP_DESCRIPTOR.DSC$A_POINTER^[1] OF
0710
0711             'L' :   IDATA[EDF$K_SURFACE_OPTION] := EDF$K_LINE_SURFACE;
0712             'F' :   IDATA[EDF$K_SURFACE_OPTION] := EDF$K_FILL_SURFACE;
0713             'A' :   IDATA[EDF$K_SURFACE_OPTION] := EDF$K_ADDED_SURFACE;
0714             'I' :   IDATA[EDF$K_SURFACE_OPTION] := EDF$K_INIT_SURFACE;
0715             'R' :   IDATA[EDF$K_SURFACE_OPTION] := EDF$K_SIZE_SURFACE;
0716             'K' :   IDATA[EDF$K_SURFACE_OPTION] := EDF$K_KEY_SURFACE;
0717
0718         OTHERWISE
0719             ( NULL-STATEMENT ) ;
0720
0721         END;    ( CASE )
0722
0723         QTAB[EDF$K_SURFACE_OPTION].DEFAULT := IDATA[EDF$K_SURFACE_OPTION];
0724
0725     END;    ( IF ODD (CL$PRESENT ('DISPLAY')) )
0726
0727     IF ODD (CL$PRESENT ('RESPONSES')) THEN
0728     BEGIN
0729         CL$GET_VALUE ('RESPONSES',TEMP_DESCRIPTOR);
0730
0731         CASE TEMP_DESCRIPTOR.DSC$A_POINTER^[1] OF
```



```
0741
0742      'A' :   [DATA[EDFSK_RESPONSES] := EDFSK_AUTO;
0743      'M' :   [DATA[EDFSK_RESPONSES] := EDFSK_MAN;
0744
0745      OTHERWISE
0746
0747      { NULL-STATEMENT } ;
0748
0749      END;      { CASE }
0750
0751      END;      { IF ODD (CLISPRESENT ('RESPONSES')) }
0752
0753      NO_INPUT      := ODD (CLISPRESENT ('CREATE'));
0754
0755      AUTO_TUNE      := FALSE;
0756
0757      END      { IF TRUE ODD (CLISPRESENT ('INTERACTIVE')) }
0758
0759      ELSE
0760      BEGIN
0761
0762      { +
0763      We don't want shifting or centering if we're nointeractive.
0764      - }
0765      FOR I := 1 TO 4 DO
0766
0767      BEGIN
0768
0769      SHIFT[I]      := NULL_CHAR;
0770      CRLF_SHIFT[I+2] := NULL_CHAR;
0771
0772      END;
0773
0774      FOR I := 1 TO 3 DO
0775
0776      LOW_SHIFT[I]      := NULL_CHAR;
0777
0778      END;
0779
0780      IF ODD (CLISPRESENT ('GRANULARITY')) THEN
0781      BEGIN
0782
0783      CLISGET_VALUE ('GRANULARITY',TEMP_DESCRIPTOR);
0784
0785      CASE TEMP_DESCRIPTOR.DSCSA_POINTER^[1] OF
0786
0787      '0','1' :   [DATA[EDFSK_GRANULARITY] := EDFSK_ONE;
0788      '2' :   [DATA[EDFSK_GRANULARITY] := EDFSK_TWO;
0789      '3' :   [DATA[EDFSK_GRANULARITY] := EDFSK_THREE;
0790      'F','4' :   [DATA[EDFSK_GRANULARITY] := EDFSK_FOUR;
0791      'D' :   [DATA[EDFSK_GRANULARITY] := EDFSK_DOUBLE;
0792
0793      'T' :   IF TEMP_DESCRIPTOR.DSCSA_POINTER^[2] = 'H' THEN
0794
0795      [DATA[EDFSK_GRANULARITY] := EDFSK_THREE
0796
0797
```

```
0798
0799         ELSE
0800             IDATA[EDFSK_GRANULARITY] := EDFSK_TWO;
0801
0802     OTHERWISE
0803         { NULL-STATEMENT } ;
0804
0805     END;    { CASE }
0806
0807 END        { IF ODD (CLISPRESENT ('GRANULARITY')) }
0808
0809 ELSE
0810     IDATA[EDFSK_GRANULARITY]      := EDFSK_THREE;
0811
0812 IF ODD (CLISPRESENT ('EMPHASIS')) THEN
0813 BEGIN
0814     CLISGET_VALUE ('EMPHASIS',TEMP_DESCRIPTOR);
0815
0816 CASE TEMP_DESCRIPTOR.DSC$A_POINTER^[1] OF
0817     'F' :      IDATA[EDFSK_BUCKET_WEIGHT] := EDFSK_FLATTER_FILES;
0818     'S' :      IDATA[EDFSK_BUCKET_WEIGHT] := EDFSK_SMALLER_BUFFERS;
0819
0820 OTHERWISE
0821     { NULL-STATEMENT } ;
0822
0823 END;    { CASE }
0824
0825 QTAB[EDFSK_BUCKET_WEIGHT].DEFAULT := IDATA[EDFSK_BUCKET_WEIGHT];
0826
0827 END        { IF ODD (CLISPRESENT ('EMPHASIS')) }
0828
0829 ELSE
0830     IDATA[EDFSK_BUCKET_WEIGHT]      := EDFSK_FLATTER_FILES;
0831
0832 DEFAULT_FILENAME_DESC := NULL_STRING;
0833 STRSTRIM (DEFAULT_FILENAME_DESC, '.FDL');
0834
0835 NL_DEV_DESC := NULL_STRING;
0836 STRSTRIM (NL_DEV_DESC, 'NL:');
0837
0838 { +
0839 Set the main loop variable to true so we can execute the main cycle.
0840 - }
0841 EDITING      := TRUE;
0842
0843 END;    { INIT_EDITOR }
```

```
0853      ( **
0854
0855      INPUT_FDL_FILE -- Uses FDL$PARSE to read the user's input FDL file.
0856
0857      This routine parses the input file using FDL$PARSE.
0858
0859      CALLING SEQUENCE:
0860
0861      INPUT_FDL_FILE:
0862
0863      INPUT PARAMETERS:
0864
0865      none
0866
0867      IMPLICIT INPUTS:
0868
0869      none
0870
0871      OUTPUT PARAMETERS:
0872
0873      none
0874
0875      IMPLICIT OUTPUTS:
0876
0877      The Definition Linked List
0878
0879      ROUTINES CALLED:
0880
0881      FDL$PARSE
0882
0883      ROUTINE VALUE:
0884
0885      none
0886
0887      SIGNALS:
0888
0889
0890      SIDE EFFECTS:
0891
0892      none
0893
0894      -- )
```



```
0896  PROCEDURE INPUT_FDL_FILE;
0897
0898  BEGIN
0899
0900      ( +
0901      Set up the condition handler for the disk.
0902      - )
0903      ESTABLISH (RMS_INPUT_COND_HANDLER);
0904
0905      ( +
0906      Now tell the user what we're doing.
0907      - )
0908      IF NOT AUTO_TUNE THEN
0909
0910          WRITELN (SHIFT,TAB,TAB,'Parsing Definition File');
0911
0912      ( +
0913      Make sure edf$line_parsed gets the non-analysis stuff.
0914      - )
0915      ANALYSIS_ONLY      := FALSE;
0916
0917      ( +
0918      Turn on the $CALLBACK flags bit to make FDL$PARSE call us.
0919      Also, turn on the SIGNAL bit to make FDL$PARSE signal errors to us.
0920      - )
0921      FLAGS.FDL$V_CALLBACK      := TRUE;
0922      FLAGS.FDL$V_SIGNAL        := TRUE;
0923
0924      ( +
0925      Parse the input file to get the old definition.
0926      If the input file doesn't exist, this will signal up to the main loop.
0927      - )
0928      ISTATUS      := FDL$PARSE (
0929                          INPUT_FILENAME_DESC,
0930                          FAB_DUMMY,
0931                          RAB_DUMMY,
0932                          FLAGS
0933                      );
0934
0935      IF (
0936      (ODD (ISTATUS))
0937      AND
0938      (NOT AUTO_TUNE)
0939      ) THEN
0940
0941          WRITELN (SHIFT,TAB,TAB,'Definition Parse Complete');
0942
0943  END;      ( INPUT_FDL_FILE )
```

```
0945 { ++
0946
0947 INPUT_ANALYSIS_FILE -- Read in the analysis file if specified.
0948
0949 This routine parses the user's analysis file if he has specified one.
0950
0951 CALLING SEQUENCE:
0952
0953 INPUT_ANALYSIS_FILE:
0954
0955 INPUT PARAMETERS:
0956
0957 none
0958
0959 IMPLICIT INPUTS:
0960
0961 none
0962
0963 OUTPUT PARAMETERS:
0964
0965 none
0966
0967 IMPLICIT OUTPUTS:
0968
0969 The Analysis Linked List
0970
0971 ROUTINES CALLED:
0972
0973 FDL$PARSE
0974
0975 ROUTINE VALUE:
0976
0977 none
0978
0979 SIGNALS:
0980
0981
0982 SIDE EFFECTS:
0983
0984 none
0985
0986 -- }
```

EDF  
V04-000

Source Listing

L 11  
16-Sep-1984 01:22:54  
5-Sep-1984 13:37:22

VAX-11 Pascal V2.4-277  
DISK\$VMSMASTER:[EDF.SRC]EDFMAIN.PAS;1 (10) Page 21

```
0988 [GLOBAL] PROCEDURE INPUT_ANALYSIS_FILE;
0989
0990 BEGIN
0991     { +
0992     Only do this if the user wants to.
0993     - }
0994     IF EDITING AND ANALYSIS_SPECIFIED THEN
0995     BEGIN
0996         { +
0997         Set up the condition handler for the disk.
0998         - }
0999         ESTABLISH (RMS_INPUT_COND_HANDLER);
1000
1001         { +
1002         Now tell the user what we're doing.
1003         - }
1004         IF NOT AUTO_TUNE THEN
1005             WRITELN (SHIFT, 'Parsing Analysis File');
1006
1007         { +
1008         Make sure edf$line_parsed gets only the analysis stuff.
1009         - }
1010         ANALYSIS_ONLY := TRUE;
1011         POINT_AT_ANALYSIS;
1012
1013         { +
1014         Create an Ident line_object and put it into the list.
1015         - }
1016         NEW_IDENT_LINE;
1017
1018         { +
1019         Turn on the $CALLBACK flags bit to make FDL$PARSE call us.
1020         Also, turn on the SIGNAL bit to make FDL$PARSE signal errors to us.
1021         - }
1022         FLAGS.FDL$V_CALLBACK := TRUE;
1023         FLAGS.FDL$V_SIGNAL := TRUE;
1024
1025         { +
1026         Parse the analysis file to get the analysis sections.
1027         If it doesn't exist, this will signal up to the main loop.
1028         - }
1029         ISTATUS := FDL$PARSE (
1030             ANALYSIS_FILENAME_DESC,
1031             FAB_DUMMY,
1032             RAB_DUMMY,
1033             FLAGS
1034         );
1035
1036         ANALYSIS_ONLY := FALSE;
1037         POINT_AT_DEFINITION;
1038
1039         IF (
1040             (ODD (ISTATUS))
1041         )
```

EDF  
V04-000

Source Listing

M 11  
16-Sep-1984 01:22:56  
5-Sep-1984 13:37:22

VAX-11 Pascal V2.4-277  
DISK\$VMSMASTER:[EDF.SRC]EDFMAIN.PAS;1 (10) Page 22

```
1045      AND  
1046      (NOT AUTO_TUNE)  
1047      ) THEN  
1048  
1049          WRITELN (SHIFT,'Analysis Parse Complete',CRLF);  
1050  
1051      END;      { IF EDITING AND ANALYSIS_SPECIFIED }  
1052  
1053      END;      { INPUT_ANALYSIS_FILE }
```



EDF  
V04-000

Source Listing

N 11  
16-Sep-1984 01:22:54  
5-Sep-1984 13:37:22

VAX-11 Pascal V2.4-277  
DISK\$VMSMASTER:[EDF.SRC]EDFMAIN.PAS;1 (11) Page 23

```
1055      ( ++
1056
1057      SETUP_CONTINUE -- Get ready to ask the user if he wants to continue after
1058      an input parse error.
1059
1060      CALLING SEQUENCE:
1061
1062      SETUP_CONTINUE:
1063
1064      INPUT PARAMETERS:
1065
1066      none
1067
1068      IMPLICIT INPUTS:
1069
1070      none
1071
1072      OUTPUT PARAMETERS:
1073
1074      none
1075
1076      IMPLICIT OUTPUTS:
1077
1078      none
1079
1080      ROUTINES CALLED:
1081
1082      CLEAR
1083
1084      ROUTINE VALUE:
1085
1086      none
1087
1088      SIGNALS:
1089
1090      none
1091
1092      SIDE EFFECTS:
1093
1094      none
1095      -- }
1096
```

EDF  
V04-000

Source Listing

0 12  
16-Sep-1984 01:22:54  
5-Sep-1984 13:37:22

VAX-11 Pascal V2.4-277  
DISK\$VMSMASTER:[EDF.SRC]EDFMAIN.PAS;1 (12) Page 24

1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109

```
PROCEDURE SETUP_CONTINUE;  
BEGIN  
    { +  
    Set up the control/Z handler in case the user types ^Z.  
    - }  
    ESTABLISH (CTRLZ_COND_HANDLER);  
    CLEAR (PAUSE);  
END;    { SETUP_CONTINUE }
```

```
1111      ( ++
1112
1113      DISPATCH_FUNCTION -- Branch off to the selected function.
1114
1115      This routine is just a big CASE statement to execute the operation the
1116      user wants.
1117
1118      CALLING SEQUENCE:
1119
1120      DISPATCH_FUNCTION
1121
1122      INPUT PARAMETERS:
1123
1124      none
1125
1126      IMPLICIT INPUTS:
1127
1128      [DATA[EDFSK_FIRST_SCRIPT]
1129      [DATA[EDFSK_CURRENT_FUNCTION]
1130
1131      OUTPUT PARAMETERS:
1132
1133      none
1134
1135      IMPLICIT OUTPUTS:
1136
1137      EDITING
1138      [DATA[EDFSK_SCRIPT_OPTION]
1139
1140      ROUTINES CALLED:
1141
1142      ADD_FDL_LINE
1143      DELETE_FDL_LINE
1144      CREATE_NEW_FDL
1145      HELP_PROC
1146      MODIFY_FDL_LINE
1147      INVOKE_SCRIPT
1148      VIEW_DEF
1149
1150      ROUTINE VALUE:
1151
1152      none
1153
1154      SIGNALS:
1155
1156
1157      SIDE EFFECTS:
1158
1159      none
1160
1161      -- )
```

```
1163 PROCEDURE DISPATCH_FUNCTION;
1164
1165 BEGIN
1166     { +
1167     Set up the control/Z handler and reinitialize some flags.
1168     - }
1169     ESTABLISH (CTRLZ_COND_HANDLER);
1170
1171     IF NOT AUTO_TUNE THEN
1172         CLOSE (FDL_DEST, ERROR := CONTINUE);
1173
1174     POINT AT DEFINITION;
1175     DEST_IS_TERMINAL := TRUE;
1176     OPTIMIZING := FALSE;
1177     VISIBLE_QUESTION := FALSE;
1178     TEMP_FUCL_PROMPT := FALSE;
1179     TAKE_DEFAULTS := AUTO_TUNE;
1180
1181     { +
1182     Ask the user only if he hadn't requested one from DCL.
1183     - }
1184     IF IDATA[EDFSK_FIRST_SCRIPT] = EDFSK_ZERO_SCRIPT THEN
1185         BEGIN
1186             { +
1187             Get the user's top-level function and dispatch on it.
1188             - }
1189             QUERY (EDFSK_CURRENT_FUNCTION);
1190
1191             CASE IDATA[EDFSK_CURRENT_FUNCTION] OF
1192                 EDFSK_ADD :      ADD_FDL_LINE; { Add a new line_object to the list. }
1193                 EDFSK_DELETE :  DELETE_FDL_LINE; { Remove a line_object from the list. }
1194                 EDFSK_HELP :    HELP_PROC; { Prompt for help and process it. }
1195                 EDFSK_INVOKE :  INVORE_SCRIPT; { Ask a bunch of related questions. }
1196                 EDFSK_MODIFY :  MODIFY_FDL_LINE; { Edit an extant line_object. }
1197                 EDFSK_QUIT :    EDITING := FALSE; { Wipe out! All bets are off! }
1198                 EDFSK_SET :     SET_PROC; { Set the editor characteristics. }
1199                 EDFSK_VIEW :    VIEW_DEF; { Show the user the definition. }
1200
1201                 EDFSK_EXIT :
1202                     BEGIN
1203                         { +
1204                         Stop the editing loop and output the new FDL file.
1205                         - }
1206                         EDITING := FALSE;
1207                         CREATE_NEW_FDL;
1208
1209                     END;
1210
1211             END;
1212
1213         END;
1214
1215     END;
1216
```



EDF  
V04-000

Source Listing

E 12  
16-Sep-1984 01:22:54  
5-Sep-1984 13:37:22

VAX-11 Pascal V2.4-277  
DISK\$VMSMASTER:[EDF.SRC]EDFMAIN.PAS;1 (15) Page 27

```
1218      OTHERWISE
1219
1220          { NULL-STATEMENT } ;
1221
1222      END;      { CASE }
1223
1224      END      { IF TRUE IDATA[EDFSK_FIRST_SCRIPT] = EDFSK_ZERO_SCRIPT }
1225
1226      ELSE
1227
1228      BEGIN
1229
1230          { +
1231          The user wants to do a script right off, do it.
1232          - }
1233          IDATA[EDFSK_SCRIPT_OPTION]      := IDATA[EDFSK_FIRST_SCRIPT];
1234
1235          ISAM_ORG      := (IDATA[EDFSK_SCRIPT_OPTION] IN [ EDFSK_ADD_KEY_FDL,
1236          EDFSK_DELETE_KEY_FDL, EDFSK_IDX_DESIGN_FDL,
1237          EDFSK_REDESIGN_FDL, EDFSK_OPTIMIZE_FDL ]);
1238
1239          INVOKE_SCRIPT;
1240
1241      END;      { IF FALSE IDATA[EDFSK_FIRST_SCRIPT] = EDFSK_ZERO_SCRIPT }
1242
1243      END;      { DISPATCH_FUNCTION }
```

```
1245 { **
1246 +-----+
1247 | *** THIS IS THE TOP LEVEL CODE IN THE UTILITY. *** |
1248 +-----+
1249 -- }
1250
1251 BEGIN
1252
1253   { +
1254   Set up the editor, setup the exit and condition handlers, a control/C
1255   AST routine, and get all the DCL switch options.
1256   Set EDITING to TRUE.
1257   - }
1258   INIT_EDITOR;
1259
1260   { +
1261   Read in the FDL file, and possibly an analysis file.
1262   1st clear the error flag.
1263   - }
1264   RMS_INPUT_ERROR      := FALSE;
1265
1266   IF NOT NO_INPUT THEN
1267     INPUT_FDL_FILE;
1268
1269   { +
1270   If we had an error, pause to let the user read the messages,
1271   otherwise, continue on.
1272   - }
1273   IF EDITING AND RMS_INPUT_ERROR THEN
1274     SETUP_CONTINUE
1275
1276   ELSE IF NOT NO_INPUT THEN
1277     LIB$WAIT (3.0);
```

EDF  
V04-000

Source Listing

6 12  
16-Sep-1984 01:22:54  
5-Sep-1984 13:37:22

VAX-11 Pascal V2.4-277  
DISK\$VMSMASTER:[EDF.SRC]EDFMAIN.PAS;1 (17) Page 29

```
1283 { +
1284 This is the Main Loop.
1285 - }
1286 WHILE EDITING DO
1287 BEGIN
1288     DISPATCH_FUNCTION;
1289     IF MAIN_CTRLZ THEN
1290     BEGIN
1291         { +
1292         Stop the editing loop and output the new FDL file.
1293         - }
1294         EDITING := FALSE;
1295         CREATE_NEW_FDL;
1296     END;
1297 END; { WHILE EDITING }
1298 { +
1299 Close the journal file if we had one.
1300 - }
1301 IF JOURNAL_ENABLED THEN
1302     CLOSE (JOURNAL_FILE);
1303 END. { EDF UTILITY. }
```

EDF  
V04-000

Generated Code

H 12  
16-Sep-1984 01:22:54  
5-Sep-1984 13:37:22

VAX-11 Pascal V2.4-277  
DISK\$VMSMASTER:[EDF.SRC]EDFMAIN.PAS;1 (17) Page 30

														.TITLE	EDF		
														.IDENT	\V04-000\		
00000														.PSECT	\$CODE,PIC,CON,REL,LCL,SHR,EXE,RD,NOWRT,2		
5F	4B	00	45	56	49	54	43	41	52	45	54	4E	49	00000	C.AAA:	.ASCII	\INTERACTIVE\<0>
		43	41	42	59	41	4C	50	24	24	46	44	45	0000C	C.AAB:	.ASCII	\EDF\$\$PLAYBACK_INPUT\<0>
49	5F	4C	41	4E	52	55	4F	4A	00	24	24	46	44	45	0001A		
															00020	C.AAC:	.ASCII
						53	49	53	59	4C	41	4E	41	0002E			
														00034	C.AAD:	.ASCII	\ANALYSIS\
						00	00	54	55	50	54	55	4F	0003C	C.AAE:	.ASCII	\ANALYSIS\
														00044	C.AAF:	.ASCII	\P1\<0><0>
		00	45	56	49	00	00	54	55	50	54	55	4F	00048	C.AAG:	.ASCII	\OUTPUT\<0><0>
						00	00	54	55	50	54	55	4F	00050	C.AAH:	.ASCII	\OUTPUT\<0><0>
		00	53	59	45	00	00	54	50	49	52	43	53	00058	C.AAI:	.ASCII	\INTERACTIVE\<0>
						00	00	54	50	49	52	43	53	00064	C.AAJ:	.ASCII	\SCRIPT\<0><0>
		00	00	00	47	00	00	54	50	49	52	43	53	0006C	C.AAK:	.ASCII	\SCRIPT\<0><0>
						00	00	54	50	49	52	43	53	00074	C.AAL:	.ASCII	\NUMBER_KEYS\<0>
		00	00	00	47	4E	49	54	50	4D	4F	52	50	00080	C.AAM:	.ASCII	\SCRIPT\<0><0>
						4E	49	54	50	4D	4F	52	50	00088	C.AAN:	.ASCII	\PROMPTING\<0><0><0>
		00	00	00	47	00	59	41	4C	50	53	49	44	00094	C.AAO:	.ASCII	\PROMPTING\<0><0><0>
						00	59	41	4C	50	53	49	44	000A0	C.AAP:	.ASCII	\DISPLAY\<0>
		00	00	00	53	00	59	41	4C	50	53	49	44	000A8	C.AAQ:	.ASCII	\DISPLAY\<0>
						45	53	4E	4F	50	53	45	52	000B0	C.AAR:	.ASCII	\RESPONSES\<0><0><0>
		00	00	00	53	45	53	4E	4F	50	53	45	52	000BC	C.AAS:	.ASCII	\RESPONSES\<0><0><0>
						00	00	45	54	41	45	52	43	000C8	C.AAT:	.ASCII	\CREATE\<0><0>
		00	59	54	49	52	41	4C	55	4E	41	52	47	000D0	C.AAU:	.ASCII	\GRANULARITY\<0>
						52	41	4C	55	4E	41	52	47	000DC	C.AAV:	.ASCII	\GRANULARITY\<0>
		00	59	54	49	53	49	53	41	48	50	4D	45	000E8	C.AAW:	.ASCII	\EMPHASIS\
						53	49	53	41	48	50	4D	45	000F0	C.AAX:	.ASCII	\EMPHASIS\
						00	00	00	00	00	00	00	00	000F8	C.AAY:	.ASCII	\.FDL\
														000FC	C.AAZ:	.ASCII	\NL:\<0>
69	6E	69	66	65	44	20	67	6E	69	73	72	61	50	00100	C.ABA:	.ASCII	\Parsing Definition File\<0>
72	61	50	20	00	65	6C	69	46	20	6E	6F	69	74	0010E			
00	00	00	65	74	65	6C	70	6D	6F	43	20	65	73	00118	C.ABB:	.ASCII	\Definition Parse Complete\<0><0><0>
73	79	6C	61	6E	41	20	67	6E	69	73	72	61	50	00126			
		00	00	00	65	00	65	6C	69	46	20	73	69	00134	C.ABC:	.ASCII	\Parsing Analysis File\<0><0><0>
						00	65	6C	69	46	20	73	69	00142			
65	73	72	61	50	20	73	69	73	79	6C	61	6E	41	0014C	C.ABD:	.ASCII	\Analysis Parse Complete\<0>
		00	65	74	65	6C	70	6D	6F	43	20	65	73	0015A			
														00164	C.ABE:	.LONG	^X67,0,0,0,0,0,0
		00000000		00000000		00000000		00000000		00000000		00000067		00178			
		00000000		00000000		00000000		00000000		00000000		00000000					



EDF  
V04-000

Generated Code

1 12  
16-Sep-1984 01:22:54  
5-Sep-1984 13:37:22

VAX-11 Pascal V2.4-277  
DISK\$VMSMASTER:[EDF.SRC]EDFMAIN.PAS;1 (17) Page 31

0000V	CF	00	FB	00047	9\$:	CALLS	#0,DISPATCH_FUNCTION	:	1290
00V00000000G	EF	00	E1	0004C		BBC	#0,MAIN_CTLRZ,12\$	:	1292
		00000000G	EF	94	00054	CLRB	EDITING	:	1299
00000000G	EF	00	FB	0005A		CALLS	#0,CREATE_NEW_FDL	:	1300
DE 00000000G	EF	00	E0	00061	12\$:	BBS	#0,EDITING,9\$	:	
00V00000000G	EF	00	E1	00069		BBC	#0,JOURNAL_ENABLED,15\$	:	1309
		00000000G	EF	9F	00071	PUSHAB	JOURNAL_FICE	:	1311
00000000G	EF	01	FB	00077		CALLS	#1,PASS\$CLOSE2	:	
	50	01	D0	0007E	15\$:	MOVL	#1,R0	:	1313
			04	00081		RET		:	

; Routine Size: 130 bytes, Routine Base: \$CODE + 00184

				00000	INIT_EDITOR:		:	0207
			003C	00000	.WORD	#M<R2,R3,R4,R5>		
			C2	00002	SUBL2	#28,SP		
			D0	00005	MOVL	#17694731,-8(FP)	:	0218
EC	AD	FFFFFDE4	0B	28	MOVC3	#11,C.AAA,-20(FP)		
			AD	9E	MOVAB	-20(FP),-4(FP)		
			AD	9F	PUSHAB	-8(FP)		
			01	FB	CALLS	#1,CLIS\$PRESENT		
			50	E8	BLBS	R0,2\$		
			01	90	MOVB	#1,TAKE_DEFAULTS	:	0226
			01	90	MOVB	#1,AUTO_TUNE	:	0227
			EF	94	CLRB	JOURNAL_ENABLED	:	0228
			01	90	MOVB	#1,QTAB+504	:	0229
			EF	D4	CLRL	IDATA+260	:	0230
			05	D0	MOVL	#5,IDATA+8	:	0231
			06	D0	MOVL	#6,QTAB+755	:	0232
			01	D0	MOVL	#1,QTAB+780	:	0233
			EF	94	CLRB	VIDEO_TERMINAL	:	0234
			EF	94	CLRB	DEC CRT	:	0235
			EF	94	CLRB	ANST CRT	:	0236
			EF	94	CLRB	REGIS	:	0237
			0000V	31	BRW	17\$		
			00	FB	CALLS	#0,EDF\$TERM_SETUP	:	0258
			50	D0	MOVL	R0,TERMINAL_SPEED	:	
			8F	DF	PUSHAL	#11763724	:	0269
			EF	9F	PUSHAB	TERMINAL_SPEED		
			02	FB	CALLS	#2,LIB\$MATCH_COND		
			50	E8	BLBS	R0,..+3		
			0000V	31	BRW	9\$		
			00	DD	PUSHL	#0	:	0273
			00	DD	PUSHL	#0		
			00	DD	PUSHL	#0		
			8F	D0	MOVL	#17694975,-8(FP)		
			EF	9E	MOVAB	TEMP_STRING255,-4(FP)		
			AD	9F	PUSHAB	-8(FP)		
			00	DD	PUSHL	#0		
			8F	D0	MOVL	#17694739,-16(FP)		
			EF	9E	MOVAB	C.AAB,-12(FP)		
			AD	9F	PUSHAB	-16(FP)		
			06	FB	CALLS	#6,SYS\$TRNLOG		
			50	D0	MOVL	R0,TEMP_STATUS		
			8F	DF	PUSHAL	#1577	:	0275
			EF	9F	PUSHAB	TEMP_STATUS		
			02	FB	CALLS	#2,LIB\$MATCH_COND		

		00V	50	E9	000EF	BLBC	R0,5\$	
			00	DD	000F2	PUSHL	#0	: 0277
			00	DD	000F4	PUSHL	#0	
			00	DD	000F6	PUSHL	#0	
		00B3800C	8F	DD	000F8	PUSHL	#11763724	
	00000000G	EF	04	FB	000FE	CALLS	#4,LIB\$STOP	
			00V	11	00105	BRB	9\$	
		00000001	8F	DF	00107	PUSHAL	#1	: 0279
		00000000G	EF	9F	0010D	PUSHAB	TEMP STATUS	
	00000000G	EF	02	FB	00113	CALLS	#2,LIB\$MATCH_COND	
		00V	50	E9	0011A	BLBC	R0,9\$	
	00000000G	EF	0B	DD	0011D	MOVL	#11,TERMINAL SPEED	: 0281
	00000000G	EF	00	FB	00124	CALLS	#0,EDF\$CTRLCAST	: 0288
		00000000G	EF	9F	0012B	PUSHAB	LINE\$PER_PAGE	: 0293
		00000000G	EF	9F	00131	PUSHAB	LINE_WIDTH	
		00000000G	EF	9F	00137	PUSHAB	DEV TYPE	
		00000000G	EF	9F	0013D	PUSHAB	SCREEN_FLAGS	
	00000000G	EF	04	FB	00143	CALLS	#4,LIB\$SCREEN_INFO	
	00000000G	EF	01	90	0014A	MOVAB	SCREEN_FLAGS,VIDEO_TERMINAL	: 0304
50	00000000G	EF	01	EF	00155	EXTZV	#6,#1,SCREEN_FLAGS,R0	: 0305
	00000000G	EF	50	90	0015E	MOVAB	R0,DEC CRT	
50	00000000G	EF	01	EF	00165	EXTZV	#1,#1,SCREEN_FLAGS,R0	: 0306
	00000000G	EF	50	90	0016E	MOVAB	R0,ANSI CRT	
50	00000000G	EF	01	EF	00175	EXTZV	#2,#1,SCREEN_FLAGS,R0	: 0307
	00000000G	EF	50	90	0017E	MOVAB	R0,REGIS	
	00000000G	EF	01	D1	00185	CMPL	LINE_WIDTH,MINIMUM_TERM_WIDTH	: 0312
		00V	19	00190	BLSS	13\$		
	00V00000000G	EF	00	E1	00192	BBC	#0,VIDEO_TERMINAL,14\$	
	00000000G	EF	01	D1	0019A	CMPL	LINE\$PER_PAGE,MINIMUM_VIDEO_PAGE	
		00V	18	001A5	BGEQ	14\$		
	00000000G	EF	DD	001A7	PUSHL	LINE\$PER_PAGE	: 0321	
	00000000G	EF	DD	001AD	PUSHL	LINE_WIDTH		
		00B38014	8F	DD	001B3	PUSHL	#2	
			04	FB	001B5	PUSHL	#11763732	
	00000000G	EF	04	FB	001BB	CALLS	#4,LIB\$SIGNAL	
		00000864	8F	DD	001C2	PUSHL	#2,148	: 0323
			07	DD	001C8	PUSHL	#7	
			04	DD	001CA	PUSHL	#4	
	00000000G	EF	9F	001CC	PUSHAB	SYSS\$OUTPUT_NAME		
			0B	DD	001D2	PUSHL	#11	
			01	DD	001D4	PUSHL	#1	
	00000000G	EF	9F	001D6	PUSHAB	PASS\$V OUTPUT		
		00000000G	07	FB	001DC	CALLS	#7,PASS\$OPEN2	
	00000000G	EF	9F	001E3	PUSHAB	PASS\$V OUTPUT	: 0324	
			01	FB	001E9	CALLS	#1,PASS\$REWRITE2	
			00	DD	001F0	PUSHL	#0	: 0331
			00	DD	001F2	PUSHL	#0	
			00	DD	001F4	PUSHL	#0	
	F8	AD	010E00FF	8F	DD	001F6	MOVL	#17694975,-8(FP)
	FC	AD	00000000G	EF	9E	001FE	MOVAB	JOURNAL_FILENAME,-4(FP)
		F8	AD	9F	00206	PUSHAB	-8(FP)	
			00	DD	00209	PUSHL	#0	
	F0	AD	010E0012	8F	DD	0020B	MOVL	#17694738,-16(FP)
	F4	AD	FFFFFFBFF	EF	9E	00213	MOVAB	C.AAC,-12(FP)
		F0	AD	9F	0021B	PUSHAB	-16(FP)	
	00000000G	EF	06	FB	0021E	CALLS	#6,SYSS\$TRNLOG	
	00000000G	EF	50	DD	00225	MOVL	R0,TEMP_STATUS	

EDF  
V04-000

Generated Code

K 12  
16-Sep-1984 01:22:54  
5-Sep-1984 13:37:22

VAX-11 Pascal V2.4-277  
DISK\$VMSMASTER:[EDF.SRC]EDFMAIN.PAS;1 (17)  
Page 33

			00000629	BF	DF	0022C	PUSHAL	#1577		: 0333
			00000000G	EF	9F	00235	PUSHAB	TEMP STATUS		
		00000000G		5C	FB	00238	CALLS	#2,LIBSMATCH_COND		
				50	90	0023F	MOVB	R0,R12		
			00000001	8F	DF	00242	PUSHAL	#1		
			00000000G	EF	9F	00248	PUSHAB	TEMP STATUS		
		00000000G		5C	FB	0024E	CALLS	#2,LIBSMATCH_COND		
00000000G	EF			50	8B	00255	BICB3	R12,R0,JOURNAL_ENABLED		
		00V00000000G		EF	00	E1	BBC	#0,JOURNAL_ENABLED,16\$		: 0339
					0C	DD	PUSHL	#12		: 0343
			000000FF	8F	DD	00267	PUSHL	#255		
				07	DD	0026D	PUSHL	#7		
				04	DD	0026F	PUSHL	#4		
			00000000G	EF	9F	00271	PUSHAB	JOURNAL_FILENAME		
			000000FF	8F	DD	00277	PUSHL	#255		
				01	DD	0027D	PUSHL	#1		
			00000000G	EF	9F	0027F	PUSHAB	JOURNAL_FILE		
		00000000G		5C	FB	00285	CALLS	#8,PASS\$OPEN2		
				EF	9F	0028C	PUSHAB	JOURNAL_FILE		: 0350
		00000000G		EF	01	FB	CALLS	#1,PASS\$REWRITE2		
						00299				
		00V00000000G		EF	00	E0	BBS	#0,DEC_CRT,20\$		: 0360
				50	01	D0	MOVL	#1,R0		: 0364
				5C	50	D0	MOVL	R0,I		
			FFFFFFFFG	EF	90	002A7	MOVB	NULL_CHAR,ANSI_RESET-1[I]		: 0368
			FFFFFFFFG	EF	90	002B3	MOVB	NULL_CHAR,ANSI_BOLD-1[I]		: 0369
			FFFFFFFFG	EF	90	002BF	MOVB	NULL_CHAR,ANSI_UNDERSCORE-1[I]		: 0370
			FFFFFFFFG	EF	90	002CB	MOVB	NULL_CHAR,ANSI_BLINK-1[I]		: 0371
			FFFFFFFFG	EF	90	002D7	MOVB	NULL_CHAR,ANSI_REVERSE-1[I]		: 0372
				BD	04	F3	AOBLEQ	#4,R0,19\$		
00000000G	EF	64		8F	00	ED	CMPZV	#0,#7,#^X64,LINE_WIDTH		: 0382
				07	00V	18	BGEQ	23\$		
			00V00000000G	EF	00	E1	BBC	#0,VIDEO_TERMINAL,23\$		
			00000001G	EF	90	002FB	MOVB	TAB,SHIFT+1		: 0386
			00000002G	EF	90	00306	MOVB	TAB,SHIFT+2		: 0387
			00000003G	EF	90	00311	MOVB	TAB,SHIFT+3		: 0388
			00000003G	EF	90	0031C	MOVB	TAB,CRLF_SHIFT+3		: 0389
			00000004G	EF	90	00327	MOVB	TAB,CRLF_SHIFT+4		: 0390
			00000005G	EF	90	00332	MOVB	TAB,CRLF_SHIFT+5		: 0391
			00V00000000G	EF	00	E1	BBC	#0,REGIS,25\$		: 0398
			00000000G	EF	90	00345	MOVB	NULL_CHAR,LOW_SHIFT		: 0402
			00000001G	EF	90	00350	MOVB	NULL_CHAR,LOW_SHIFT+1		: 0403
			00000002G	EF	90	00358	MOVB	NULL_CHAR,LOW_SHIFT+2		: 0404
				00V	11	00366	BRB	26\$		
			00000000G	EF	90	00368	MOVB	SHIFT+1,LOW_SHIFT		: 0412
			00000001G	EF	90	00373	MOVB	SHIFT+2,LOW_SHIFT+1		: 0413
			00000002G	EF	90	0037E	MOVB	SHIFT+3,LOW_SHIFT+2		: 0414
				8F	DF	00389	PUSHAL	#3		: 0421
			00000000G	EF	01	FB	CALLS	#1,CLEAR		
			00000000G	EF	08	D0	MOVL	#8,PARAM_BLOCK		: 0430
00000000G	EF	01		21	F0	0039D	INSV	#1,#33,#T,PARAM_BLOCK		: 0431
				50	9E	003A6	MOVAB	FDSLAL_BLOCK,R0		: 0440
		00000000G		50	C1	003AD	ADDL3	FDSLAL_BLOCK,R0,FDL_BLOCK		
				50	D0	003B9	MOVL	FDL_BLOCK,R0		: 0446
			04	A0	9E	003C0	MOVAB	EDF\$LINE_PARSED,4(R0)		
			00000104G	EF	01	D0	MOVL	#1,IDATA\$260		: 0451
			00000011G	EF	01	90	MOVB	#1,BDATA+17		: 0452



00000098G	EF	000000000G	01	DO	003D6	MOVL	#1, IDATA+152	0453
000002CDG	EF	000000000G	EF	9E	003DD	MOVAB	EDF\$AB_CARR_TABLE_KEY, QTAB+717	0458
000002D1G	EF	000000000G	EF	9E	003E8	MOVAB	EDF\$AB_CARR_TABLE_STA, QTAB+721	0459
0000053EG	EF	000000000G	EF	9E	003F3	MOVAB	EDF\$AB_FORMAT_TABLE_KEY, QTAB+1342	0460
00000542G	EF	000000000G	EF	9E	003FE	MOVAB	EDF\$AB_FORMAT_TABLE_STA, QTAB+1346	0461
0000045DG	EF	000000000G	EF	9E	00409	MOVAB	EDF\$AB_TYPE_TABLE_KEY, QTAB+1117	0462
00000461G	EF	000000000G	EF	9E	00414	MOVAB	EDF\$AB_TYPE_TABLE_STA, QTAB+1121	0463
00000476G	EF	000000000G	EF	9E	0041F	MOVAB	EDF\$AB_LOAD_METHOD_TABLE_KEY, QTAB+1142	0464
0000047AG	EF	000000000G	EF	9E	0042A	MOVAB	EDF\$AB_LOAD_METHOD_TABLE_STA, QTAB+1146	0465
000002B4G	EF	000000000G	EF	9E	00435	MOVAB	EDF\$AB_WEIGHT_TABLE_KEY, QTAB+692	0466
000002B8G	EF	000000000G	EF	9E	00440	MOVAB	EDF\$AB_WEIGHT_TABLE_STA, QTAB+696	0467
000005D4G	EF	000000000G	EF	9E	0044B	MOVAB	EDF\$AB_SURFACE_OPTION_TABLE_KEY, QTAB+1492	0468
000005D8G	EF	000000000G	EF	9E	00456	MOVAB	EDF\$AB_SURFACE_OPTION_TABLE_STA, QTAB+1496	0469
000002FFG	EF	000000000G	EF	9E	00461	MOVAB	EDF\$AB_CURRENT_FUNC_TABLE_KEY, QTAB+767	0470
00000303G	EF	000000000G	EF	9E	0046C	MOVAB	EDF\$AB_CURRENT_FUNC_TABLE_STA, QTAB+771	0471
00000318G	EF	000000000G	EF	9E	00477	MOVAB	EDF\$AB_DESIGN_CYCLE_TABLE_KEY, QTAB+792	0472
0000031CG	EF	000000000G	EF	9E	00482	MOVAB	EDF\$AB_DESIGN_CYCLE_TABLE_STA, QTAB+796	0473
00000570G	EF	000000000G	EF	9E	0048D	MOVAB	EDF\$AB_SCRIPT_OPTION_TABLE_KEY, QTAB+1392	0474
00000574G	EF	000000000G	EF	9E	00498	MOVAB	EDF\$AB_SCRIPT_OPTION_TABLE_STA, QTAB+1396	0475
000000D9G	EF	000000000G	EF	9E	004A3	MOVAB	EDF\$AB_YES_NO_TABLE_KEY, QTAB+217	0476
000000DDG	EF	000000000G	EF	9E	004AE	MOVAB	EDF\$AB_YES_NO_TABLE_STA, QTAB+221	0477
000000F2G	EF	000000000G	EF	9E	004B9	MOVAB	EDF\$AB_YES_NO_TABLE_KEY, QTAB+242	0478
000000F6G	EF	000000000G	EF	9E	004C4	MOVAB	EDF\$AB_YES_NO_TABLE_STA, QTAB+246	0479
0000010BG	EF	000000000G	EF	9E	004CF	MOVAB	EDF\$AB_YES_NO_TABLE_KEY, QTAB+267	0480
0000010FG	EF	000000000G	EF	9E	004DA	MOVAB	EDF\$AB_YES_NO_TABLE_STA, QTAB+271	0481
0000016FG	EF	000000000G	EF	9E	004E5	MOVAB	EDF\$AB_YES_NO_TABLE_KEY, QTAB+367	0482
00000173G	EF	000000000G	EF	9E	004F0	MOVAB	EDF\$AB_YES_NO_TABLE_STA, QTAB+371	0483
00000156G	EF	000000000G	EF	9E	004FB	MOVAB	EDF\$AB_YES_NO_TABLE_KEY, QTAB+342	0484
0000015AG	EF	000000000G	EF	9E	00506	MOVAB	EDF\$AB_YES_NO_TABLE_STA, QTAB+346	0485
00000124G	EF	000000000G	EF	9E	00511	MOVAB	EDF\$AB_YES_NO_TABLE_KEY, QTAB+292	0486
00000128G	EF	000000000G	EF	9E	0051C	MOVAB	EDF\$AB_YES_NO_TABLE_STA, QTAB+296	0487
0000013DG	EF	000000000G	EF	9E	00527	MOVAB	EDF\$AB_YES_NO_TABLE_KEY, QTAB+317	0488
00000141G	EF	000000000G	EF	9E	00532	MOVAB	EDF\$AB_YES_NO_TABLE_STA, QTAB+321	0489
00000205G	EF	000000000G	EF	9E	0053D	MOVAB	EDF\$AB_YES_NO_TABLE_KEY, QTAB+517	0490
00000209G	EF	000000000G	EF	9E	00548	MOVAB	EDF\$AB_YES_NO_TABLE_STA, QTAB+521	0491
000001ECG	EF	000000000G	EF	9E	00553	MOVAB	EDF\$AB_YES_NO_TABLE_KEY, QTAB+492	0492
000001F0G	EF	000000000G	EF	9E	0055E	MOVAB	EDF\$AB_YES_NO_TABLE_STA, QTAB+496	0493
000001D3G	EF	000000000G	EF	9E	00569	MOVAB	EDF\$AB_YES_NO_TABLE_KEY, QTAB+467	0494
000001D7G	EF	000000000G	EF	9E	00574	MOVAB	EDF\$AB_YES_NO_TABLE_STA, QTAB+471	0495
000001BAG	EF	000000000G	EF	9E	0057F	MOVAB	EDF\$AB_YES_NO_TABLE_KEY, QTAB+442	0496
000001BEG	EF	000000000G	EF	9E	0058A	MOVAB	EDF\$AB_YES_NO_TABLE_STA, QTAB+446	0497
00000188G	EF	000000000G	EF	9E	00595	MOVAB	EDF\$AB_YES_NO_TABLE_KEY, QTAB+392	0498
0000018CG	EF	000000000G	EF	9E	005A0	MOVAB	EDF\$AB_YES_NO_TABLE_STA, QTAB+396	0499
000001A1G	EF	000000000G	EF	9E	005AB	MOVAB	EDF\$AB_YES_NO_TABLE_KEY, QTAB+417	0500
000001A5G	EF	000000000G	EF	9E	005B6	MOVAB	EDF\$AB_YES_NO_TABLE_STA, QTAB+421	0501
000005EDG	EF	000000000G	EF	9E	005C1	MOVAB	EDF\$AB_PRIMARY_TABLE_KEY, QTAB+1517	0502
000005F1G	EF	000000000G	EF	9E	005CC	MOVAB	EDF\$AB_PRIMARY_TABLE_STA, QTAB+1521	0503
00000589G	EF	000000000G	EF	9E	005D7	MOVAB	EDF\$AB_SET_FUNCTION_TABLE_KEY, QTAB+1417	0504
0000058DG	EF	000000000G	EF	9E	005E2	MOVAB	EDF\$AB_SET_FUNCTION_TABLE_STA, QTAB+1421	0505
00000395G	EF	000000000G	EF	9E	005ED	MOVAB	EDF\$AB_GRANULARITY_TABLE_KEY, QTAB+917	0506
00000399G	EF	000000000G	EF	9E	005F8	MOVAB	EDF\$AB_GRANULARITY_TABLE_STA, QTAB+921	0507
00000525G	EF	000000000G	EF	9E	00603	MOVAB	EDF\$AB_PROMPTING_TABLE_KEY, QTAB+1317	0508
00000529G	EF	000000000G	EF	9E	0060E	MOVAB	EDF\$AB_PROMPTING_TABLE_STA, QTAB+1321	0509
00000557G	EF	000000000G	EF	9E	00619	MOVAB	EDF\$AB_RESPONSES_TABLE_KEY, QTAB+1367	0510
0000055BG	EF	000000000G	EF	9E	00624	MOVAB	EDF\$AB_RESPONSES_TABLE_STA, QTAB+1371	0511
00000000G	EF		00	FB	0062F	CALLS	#0, NEW IDENT LINE	0516
F8	AD	010E0008	8F	DO	00636	MOVL	#17694728, -8(TFP)	0521



Generated Code									
FO	AD	FFFFF7E7	EF	08	28	0063E	MOV	C	AAD,-16(FP)
		FC	AD	AD	9E	00647	MOV	AB	-16(FP),-4(FP)
			F8	AD	9F	0064C	PUSH	AB	-8(FP)
00000000G	EF	00000000G	50	01	FB	0064F	CALL		#1,CLISPRESENT
	00V	00000000G	EF	8F	8B	00656	BIC	B3	#XFE,RO,ANALYSIS SPECIFIED
	00000000G	EF	00000000G	00	E1	0065F	BBC		#0,ANALYSIS SPECIFIED,29\$
			00000000G	EF	7D	00667	MOV	Q	NULL STRING,ANALYSIS_FILENAME_DESC
			010E0008	EF	9F	00672	PUSH	AB	ANALYSIS_FILENAME_DESC
FO	AD	FFFFF7AD	EF	8F	DU	00678	MOVL		#17694728,-8(FP)
		FC	AD	08	28	00680	MOV	C3	#8,C.AAE,-16(FP)
			F8	AD	9E	00689	MOV	AB	-16(FP),-4(FP)
				AD	9F	0068E	PUSH	AB	-8(FP)
		00000000G	EF	02	FB	00691	CALL		#2,CLISGET_VALUE
		00000000G	EF	7D	00698	29\$:	MOV	Q	NULL STRING,INPUT_FILENAME_DESC
			00000000G	EF	9F	006A3	PUSH	AB	INPUT_FILENAME_DESC
			010E0002	8F	DO	006A9	MOVL		#17694722,-8(FP)
		F8	AD	EF	BO	006B1	MOV	W	C.AAF,-12(FP)
		F4	AD	AD	9E	006B9	MOV	AB	-12(FP),-4(FP)
		FC	AD	AD	9F	006BE	PUSH	AB	-8(FP)
				02	FB	006C1	CALL		#2,CLISGET_VALUE
		00000000G	EF	7D	006C8		MOV	Q	NULL STRING,OUTPUT_FILENAME_DESC
		00000000G	EF	8F	DO	006D3	MOVL		#17694726,-8(FP)
FO	AD	FFFFF75E	EF	06	28	006DB	MOV	C3	#6,C.AAG,-16(FP)
		FC	AD	AD	9E	006E4	MOV	AB	-16(FP),-4(FP)
			F8	AD	9F	006E9	PUSH	AB	-8(FP)
		00000000G	EF	01	FB	006EC	CALL		#1,CLISPRESENT
			00V	50	E9	006F3	BLBC		RO,33\$
			00000000G	EF	9F	006F6	PUSH	AB	OUTPUT_FILENAME_DESC
			010E0006	8F	DO	006FC	MOVL		#17694726,-8(FP)
FO	AD	FFFFF73D	EF	06	28	00704	MOV	C3	#6,C.AAH,-16(FP)
		FC	AD	AD	9E	0070D	MOV	AB	-16(FP),-4(FP)
			F8	AD	9F	00712	PUSH	AB	-8(FP)
		00000000G	EF	02	FB	00715	CALL		#2,CLISGET_VALUE
				00V	11	0071C	BRB		35\$
			00000000G	EF	9F	0071E	33\$:	PUSH	AB
			00000000G	EF	9F	00724		PUSH	AB
		00000000G	EF	02	FB	0072A	CALL		#2,LIB\$SCOPY_DXDX
		F8	AD	8F	DO	00731	MOVL		#17694731,-8(FP)
EC	AD	FFFFF710	EF	0B	28	00739	MOV	C3	#11,C.AAI,-20(FP)
		FC	AD	AD	9E	00742	MOV	AB	-20(FP),-4(FP)
			F8	AD	9F	00747	PUSH	AB	-8(FP)
		00000000G	EF	01	FB	0074A	CALL		#1,CLISPRESENT
			03	50	E8	00751	BLBS		RO,+3
				0000V	31	00754	BRW		87\$
		F8	AD	8F	DO	00757	MOVL		#17694726,-8(FP)
FO	AD	FFFFF6F6	EF	06	28	0075F	MOV	C3	#6,C.AAJ,-16(FP)
		FC	AD	AD	9E	00768	MOV	AB	-16(FP),-4(FP)
			F8	AD	9F	0076D	PUSH	AB	-8(FP)
		00000000G	EF	01	FB	00770	CALL		#1,CLISPRESENT
			03	50	E8	00777	BLBS		RO,+3
				0000V	31	0077A	BRW		48\$
		F8	AD	EF	7D	0077D	MOV	Q	NULL STRING,TEMP_DESCRIPTOR
			F8	AD	9F	00785	PUSH	AB	TEMP_DESCRIPTOR
			010E0006	8F	DO	00788	MOVL		#17694726,-16(FP)
E8	AD	FFFFF6CD	EF	06	28	00790	MOV	C3	#6,C.AAK,-24(FP)
		F4	AD	AD	9E	00799	MOV	AB	-24(FP),-12(FP)
			F8	AD	9F	0079E	PUSH	AB	-16(FP)

00000000G	EF	02	FB	007A1	CALLS	#2,CLISGET_VALUE	
13	41	8F	FC	8D	MOVZBL	TEMP_DESCRIPTOR+4,R0	: 0586
				50	CASEB	R0,#65,#19	
		0000V		8F	.DISPL	39\$	
		0028		007B1	.DISPL	40	
		0028		007B3	.DISPL	40	
		0028		007B5	.DISPL	40	
		0000V		007B7	.DISPL	40\$	
		0028		007B9	.DISPL	40	
		0028		007BB	.DISPL	40	
		0028		007BD	.DISPL	40	
		0028		007BF	.DISPL	40	
		0000V		007C1	.DISPL	41\$	
		0028		007C3	.DISPL	40	
		0028		007C5	.DISPL	40	
		0028		007C7	.DISPL	40	
		0028		007C9	.DISPL	40	
		0028		007CB	.DISPL	40	
		0000V		007CD	.DISPL	43\$	
		0028		007CF	.DISPL	40	
		0028		007D1	.DISPL	40	
		0000V		007D3	.DISPL	44\$	
		0000V		007D5	.DISPL	42\$	
		0000V		007D7	.DISPL	45\$	
		00V	11	007D9	BRB	46\$	
		00000008G	EF	D4	CLRL	IDATA+8	: 0588
			00V	11	BRB	47\$	
00000008G	EF	01	D0	007E3	MOVL	#1,IDATA+8	: 0590
		00V	11	007EA	BRB	47\$	
00000008G	EF	02	D0	007EC	MOVL	#2,IDATA+8	: 0592
		00V	11	007F3	BRB	47\$	
00000008G	EF	04	D0	007F5	MOVL	#4,IDATA+8	: 0594
		00V	11	007FC	BRB	47\$	
00000008G	EF	05	D0	007FE	MOVL	#5,IDATA+8	: 0596
		00V	11	00805	BRB	47\$	
00000008G	EF	03	D0	00807	MOVL	#3,IDATA+8	: 0598
		00V	11	0080E	BRB	47\$	
00000008G	EF	06	D0	00810	MOVL	#6,IDATA+8	: 0600
		00V	11	00817	BRB	47\$	
00000008G	EF	07	D0	00819	MOVL	#7,IDATA+8	: 0607
		F8	AD	9F	PUSHAB	TEMP_DESCRIPTOR	: 0611
00000000G	EF	01	FB	00823	CALLS	#1,STR\$FREE1_DX	
		00V	11	0082A	BRB	49\$	
00000008G	EF	07	D0	0082C	MOVL	#7,IDATA+8	: 0617
	F0	AD	010E000B	8F	MOVL	#17694731,-16(FP)	: 0622
E4	AD	FFFFF62A	EF	0B	MOVC3	#11,C.AAL,-28(FP)	
		F4	AD	AD	MOVAB	-28(FP),-12(FP)	
			F0	AD	PUSHAB	-16(FP)	
00000000G	EF	01	FB	0084C	CALLS	#1,CLISPRESENT	
	00V	50	E9	00853	BLBC	R0,52\$	
	F8	AD	00000000G	EF	MOVQ	NULL_STRING,TEMP_DESCRIPTOR	: 0626
		F8	AD	9F	PUSHAB	TEMP_DESCRIPTOR	: 0627
		F0	AD	010E0006	8F	MOVL	#17694726,-16(FP)
E8	AD	FFFFF608	EF	06	MOVC3	#6,C.AAM,-24(FP)	
		F4	AD	AD	MOVAB	-24(FP),-12(FP)	
			F0	AD	PUSHAB	-16(FP)	
00000000G	EF	02	FB	0087A	CALLS	#2,CLISGET_VALUE	
		00000F0G	EF	9F	PUSHAB	IDATA+240	: 0628

			F8	AD	9F	00887	PUSHAB	TEMP_DESCRIPTOR	
	00000000G	EF		02	FB	0088A	CALLS	#2,OTS\$CVT_T1_L	
	00000000G	EF		50	D0	00891	MOVL	R0,ISTATUS	
	000004CEG	EF	000000F0G	EF	D0	00898	MOVL	IDATA+240,QTAB+1230	: 0630
	00000000G	EF		01	90	008A3	MOVB	#1,NUMBER_KEYS_SET	: 0631
			F8	AD	9F	008AA	PUSHAB	TEMP_DESCRIPTOR	: 0633
	00000000G	EF		01	FB	008AD	CALLS	#1,STR\$FREE1_DX	
E4	AD FFFFF5BD	EF	010E0009	8F	D0	008B4	MOVL	#17694729,-16(FP)	: 0641
	F4	AD		09	28	008BC	MOVC3	#9,C.AAN,-28(FP)	
			E4	AD	9E	008C5	MOVAB	-28(FP),-12(FP)	
			F0	AD	9F	008CA	PUSHAB	-16(FP)	
	00000000G	EF		01	FB	008CD	CALLS	#1,CLIS\$PRESENT	
		00V		50	E9	008D4	BLBC	R0,63\$	
	F8	AD	00000000G	EF	7D	008D7	MOVQ	NULL_STRING,TEMP_DESCRIPTOR	: 0648
			F8	AD	9F	008DF	PUSHAB	TEMP_DESCRIPTOR	: 0649
E4	AD FFFFF59B	EF	010E0009	8F	D0	008E2	MOVL	#17694729,-16(FP)	
	F4	AD		09	28	008EA	MOVC3	#9,C.AAO,-28(FP)	
			E4	AD	9E	008F3	MOVAB	-28(FP),-12(FP)	
			F0	AD	9F	008F8	PUSHAB	-16(FP)	
	00000000G	EF		02	FB	008FB	CALLS	#2,CLIS\$GET_VALUE	
		50		BD	9A	00902	MOVZBL	@TEMP_DESCRIPTOR+4,R0	: 0654
04	42	8F	FC	50	8F	00906	CASEB	R0,#68,#4	
				0000V		00908	.DISPL	55\$	
				000A		0090D	.DISPL	10	
				000A		0090F	.DISPL	10	
				000A		00911	.DISPL	10	
				0000V		00913	.DISPL	56\$	
				00V	11	00915	BRB	57\$	
			00000000G	EF	94	00917	CLRB	FULL_PROMPT	: 0661
				00V	11	0091D	BRB	62\$	
	00000000G	EF		01	90	0091F	MOVB	#1,FULL_PROMPT	: 0668
				00V	11	00926	BRB	62\$	
00V00000000G	EF			00	E1	00928	BBC	#0,VIDEO_TERMINAL,60\$	: 0676
	0B	00000000G		EF	D1	00930	CMPL	TERMINAL_SPEED,#11	
				00V	19	00937	BLSS	60\$	
	00000000G	EF		01	90	00939	MOVB	#1,FULL_PROMPT	: 0680
				00V	11	00940	BRB	62\$	
			00000000G	EF	94	00942	CLRB	FULL_PROMPT	: 0684
			F8	AD	9F	00948	PUSHAB	TEMP_DESCRIPTOR	: 0688
	00000000G	EF		01	FB	0094B	CALLS	#1,STR\$FREE1_DX	
		00V		11	00952	BRB	68\$		
00V00000000G	EF			00	E1	00954	BBC	#0,VIDEO_TERMINAL,66\$	: 0699
	0B	00000000G		EF	D1	0095C	CMPL	TERMINAL_SPEED,#11	
				00V	19	00963	BLSS	66\$	
	00000000G	EF		01	90	00965	MOVB	#1,FULL_PROMPT	: 0701
				00V	11	0096C	BRB	68\$	
			00000000G	EF	94	0096E	CLRB	FULL_PROMPT	: 0705
E8	AD FFFFF515	EF	010E0007	8F	D0	00974	MOVL	#17694727,-16(FP)	: 0709
	F4	AD		07	28	0097C	MOVC3	#7,C.AAP,-24(FP)	
			E8	AD	9E	00985	MOVAB	-24(FP),-12(FP)	
			F0	AD	9F	0098A	PUSHAB	-16(FP)	
	00000000G	EF		01	FB	0098D	CALLS	#1,CLIS\$PRESENT	
		03		50	E8	00994	BLBS	R0,+3	
				0000V	31	00997	BRW	79\$	
			F8	AD	9F	0099A	PUSHAB	TEMP_DESCRIPTOR	: 0713
E8	AD FFFFF4F4	EF	010E0007	8F	D0	0099D	MOVL	#17694727,-16(FP)	
				07	28	009A5	MOVC3	#7,C.AAQ,-24(FP)	

	F4	AD	E8	AD	9E	009AE	MOVAB	-24(FP),-12(FP)	
			F0	AD	9F	009B3	PUSHAB	-16(FP)	
	00000000G	EF		02	FB	009B6	CALLS	#2,CLISGET VALUE	
		50	FC	BD	9A	009BD	MOVZBL	@TEMP_DESCRIPTOR+4,R0	: 0715
11	41	8F		50	8F	009C1	CASEB	R0,#65,#17	
				0000V		009C6	.DISPL	73\$	
				0024		009C8	.DISPL	36	
				0024		009CA	.DISPL	36	
				0024		009CC	.DISPL	36	
				0024		009CE	.DISPL	36	
				0000V		009D0	.DISPL	72\$	
				0024		009D2	.DISPL	36	
				0024		009D4	.DISPL	36	
				0000V		009D6	.DISPL	74\$	
				0024		009D8	.DISPL	36	
				0000V		009DA	.DISPL	76\$	
				0000V		009DC	.DISPL	71\$	
				0024		009DE	.DISPL	36	
				0024		009E0	.DISPL	36	
				0024		009E2	.DISPL	36	
				0024		009E4	.DISPL	36	
				0024		009E6	.DISPL	36	
				0000V		009E8	.DISPL	75\$	
				00V	11	009EA	BRB	77\$	
	00000118G	EF		05	D0	009EC	MOVL	#5, IDATA+280	: 0717
				00V	11	009F3	BRB	78\$	
				00000118G	EF	D4	009F5	72\$: CLRL	IDATA+280 : 0718
				00V	11	009FB	BRB	78\$	
	00000118G	EF		03	D0	009FD	73\$: MOVL	#3, IDATA+280	: 0719
				00V	11	00A04	BRB	78\$	
	00000118G	EF		02	D0	00A06	74\$: MOVL	#2, IDATA+280	: 0720
				00V	11	00A0D	BRB	78\$	
	00000118G	EF		01	D0	00A0F	75\$: MOVL	#1, IDATA+280	: 0721
				00V	11	00A16	BRB	78\$	
	00000118G	EF		04	D0	00A18	76\$: MOVL	#4, IDATA+280	: 0722
				00V	11	00A1F	BRB	78\$	
						00A21	77\$: MOVL	IDATA+280,QTAB+1480	: 0730
	000005C8G	EF	00000118G	EF	D0	00A21	78\$: MOVL	#17694729,-16(FP)	: 0734
	F0	AD	010E0009	8F	D0	00A2C	79\$: MOVL	#9,C,AAR,-28(FP)	
E4	AD	FFFFF46D	EF	09	28	00A34	MOVC3	-28(FP),-12(FP)	
	F4	AD		E4	AD	9E	00A3D	MOVAB	-16(FP)
				F0	AD	9F	00A42	PUSHAB	-16(FP)
	00000000G	EF		01	FB	00A45	CALLS	#1,CLISPRESENT	
		00V		50	E9	00A4C	BLBC	R0,86\$	
				F8	AD	9F	00A4F	PUSHAB	TEMP_DESCRIPTOR
				8F	D0	00A52	MOVL	#17694729,-16(FP)	: 0738
E4	AD	FFFFF453	EF	09	28	00A5A	MOVC3	#9,C,AAS,-28(FP)	
	F4	AD		E4	AD	9E	00A63	MOVAB	-28(FP),-12(FP)
				F0	AD	9F	00A68	PUSHAB	-16(FP)
	00000000G	EF		02	FB	00A6B	CALLS	#2,CLISGET VALUE	
		50		FC	BD	9A	00A72	MOVZBL	@TEMP_DESCRIPTOR+4,R0
0C	41	8F		50	8F	00A76	CASEB	R0,#65,#12	: 0740
				0000V		00A7B	.DISPL	82\$	
				001A		00A7D	.DISPL	26	
				001A		00A7F	.DISPL	26	
				001A		00A81	.DISPL	26	
				001A		00A83	.DISPL	26	



				001A	00A85	.DISPL	26	
				001A	00A87	.DISPL	26	
				001A	00A89	.DISPL	26	
				001A	00A8B	.DISPL	26	
				001A	00A8D	.DISPL	26	
				001A	00A8F	.DISPL	26	
				001A	00A91	.DISPL	26	
				00COV	00A93	.DISPL	83\$	
				00V	11 00A95	BRB	84\$	
		00000104G		EF	D4 00A97	CLRL	IDATA+260	: 0742
				00V	11 00A9D	BRB	86\$	
	00000104G	EF		01	D0 00A9F	MOVL	#1, IDATA+260	: 0743
				00V	11 00AA6	BRB	86\$	
					00AA8	84\$:		
					00AA8	86\$:		
					00AB0	MOVL	#17694726, -16(FP)	: 0753
					00AB0	MOV C3	#6, C.AAT, -24(FP)	
					00AB9	MOVAB	-24(FP), -12(FP)	
					00ABE	PUSHAB	-16(FP)	
					00AC1	CALLS	#1, CLIS\$PRESENT	
					00AC8	BICB3	#^XFE, R0, NO_INPUT	
					00AD1	CLRB	AUTO_TUNE	: 0755
					00AD7	BRB	90\$	
					00AD9	87\$:		
					00ADC	88\$:		
					00ADF	MOVL	R0, I	: 0766
					00AEB	MOVB	NULL_CHAR, SHIFT-1[I]	: 0770
					00AF7	MOVB	NULL_CHAR, CRLF_SHIFT+1[I]	: 0771
					00AFB	AOBLEQ	#4, R0, 88\$	
					00AFE	MOVL	#1, R0	: 0775
					00B01	MOVL	R0, I	
					00B0D	89\$:		
					00B11	MOVB	NULL_CHAR, LOW_SHIFT-1[I]	: 0777
					00B19	AOBLEQ	#3, R0, 89\$	
					00B22	MOVL	#17694731, -16(FP)	: 0781
					00B27	MOV C3	#11, C.AAU, -28(FP)	
					00B2A	MOVAB	-28(FP), -12(FP)	
					00B31	PUSHAB	-16(FP)	
					00B34	CALLS	#1, CLIS\$PRESENT	
					00B37	BLBS	R0, +3	
					00B3A	BRW	104\$	
					00B42	PUSHAB	TEMP_DESCRIPTOR	: 0785
					00B48	MOVL	#17694731, -16(FP)	
					00B50	MOV C3	#11, C.AAV, -28(FP)	
					00B53	MOVAB	-28(FP), -12(FP)	
					00B5A	PUSHAB	-16(FP)	
					00B5E	CALLS	#2, CLIS\$GET_VALUE	
					00B62	MOVZBL	@TEMP_DESCRIPTOR+4, R0	: 0787
					00B64	CASEB	R0, #49, #35	
					00B66	.DISPL	93\$	
					00B68	.DISPL	94\$	
					00B6A	.DISPL	95\$	
					00B6C	.DISPL	96\$	
					00B6E	.DISPL	72	
					00B70	.DISPL	72	
					00B72	.DISPL	72	
					00B74	.DISPL	72	
					00B76	.DISPL	72	

			0048	00B78	.DISPL	72		
			0048	00B7A	.DISPL	72		
			0048	00B7C	.DISPL	72		
			0048	00B7E	.DISPL	72		
			0048	00B80	.DISPL	72		
			0048	00B82	.DISPL	72		
			0048	00B84	.DISPL	72		
			0048	00B86	.DISPL	72		
			0000V	00B88	.DISPL	97\$		
			0048	00B8A	.DISPL	72		
			0000V	00B8C	.DISPL	96\$		
			0048	00B8E	.DISPL	72		
			0048	00B90	.DISPL	72		
			0048	00B92	.DISPL	72		
			0048	00B94	.DISPL	72		
			0048	00B96	.DISPL	72		
			0048	00B98	.DISPL	72		
			0048	00B9A	.DISPL	72		
			0048	00B9C	.DISPL	72		
			0000V	00B9E	.DISPL	93\$		
			0048	00BA0	.DISPL	72		
			0048	00BA2	.DISPL	72		
			0048	00BA4	.DISPL	72		
			0048	00BA6	.DISPL	72		
			0000V	00BA8	.DISPL	98\$		
			0000V	31 00BAA	BRW	102\$		
		000000BCG	EF	D4 00BAD	93\$:	CLRL	IDATA+188 : 0789	
			00V	11 00BB3		BRB	105\$	
	000000BCG	EF	01	D0 00BB5	94\$:	MOVL	#1, IDATA+188 : 0790	
			00V	11 00BBC		BRB	105\$	
	000000BCG	EF	02	D0 00BBE	95\$:	MOVL	#2, IDATA+188 : 0791	
			00V	11 00BC5		BRB	105\$	
	000000BCG	EF	03	D0 00BC7	96\$:	MOVL	#3, IDATA+188 : 0792	
			00V	11 00BCE		BRB	105\$	
	000000BCG	EF	04	D0 00BD0	97\$:	MOVL	#4, IDATA+188 : 0793	
			00V	11 00BD7		BRB	105\$	
		50	AD	D0 00BD9	98\$:	MOVL	TEMP_DESCRIPTOR+4, R0 : 0795	
	48	8F	FC	A0 91 00BDD		CMPB	1(R0), #72	
			01	00V	12 00BE2	BNEQ	100\$	
	000000BCG	EF	02	D0 00BE4		MOVL	#2, IDATA+188 : 0797	
			00V	11 00BEB		BRB	105\$	
	000000BCG	EF	01	D0 00BED	100\$:	MOVL	#1, IDATA+188 : 0801	
			00V	11 00BF4		BRB	105\$	
			00V	11 00BF6	102\$:	BRB	105\$	
	000000BCG	EF	02	D0 00BF8	104\$:	MOVL	#2, IDATA+188 : 0813	
		F0	AD	010E0008	8F	D0 00BFF	105\$:	
E8	AD	FFFFFF2D2	EF	08	28 00C07	MOVC3	#8, C, AAW, -24(FP) : 0815	
		F4	AD		AD 9E 00C10	MOVAB	-24(FP), -12(FP)	
					AD 9F 00C15	PUSHAB	-16(FP)	
	00000000G	EF	01	FB 00C18	CALLS	#1, CLISPRESNT		
		00V	50	E9 00C1F	BLBC	R0, 112\$		
			FB	AD 9F 00C22	PUSHAB	TEMP_DESCRIPTOR	: 0819	
		F0	AD	010E0008	8F	D0 00C25	MOVL	#17694728, -16(FP)
E8	AD	FFFFFF2B4	EF	08	28 00C2D	MOVC3	#8, C, AAX, -24(FP)	
		F4	AD		AD 9E 00C36	MOVAB	-24(FP), -12(FP)	
					AD 9F 00C3B	PUSHAB	-16(FP)	
	00000000G	EF	02	FB 00C3E	CALLS	#2, CLISGET_VALUE		

0D	46	50	FC	BD	9A	00C45	MOVZBL	@TEMP_DESCRIPTOR+4,R0	: 0821
		8F		50	8F	00C49	CASEB	R0,#70,#13	
				0000V		00C4E	.DISPL	108\$	
				001C		00C50	.DISPL	28	
				001C		00C52	.DISPL	28	
				001C		00C54	.DISPL	28	
				001C		00C56	.DISPL	28	
				001C		00C58	.DISPL	28	
				001C		00C5A	.DISPL	28	
				001C		00C5C	.DISPL	28	
				001C		00C5E	.DISPL	28	
				001C		00C60	.DISPL	28	
				001C		00C62	.DISPL	28	
				001C		00C64	.DISPL	28	
				001C		00C66	.DISPL	28	
				0000V		00C68	.DISPL	109\$	
00000098G	EF			00V	11	00C6A	BRB	110\$	
				01	D0	00C6C	MOVL	#1, IDATA+152	: 0823
				00V	11	00C73	BRB	111\$	
		00000098G	EF	D4	00C75	109\$:	CLRL	IDATA+152	: 0824
				00V	11	00C7B	BRB	111\$	
						00C7D		110\$:	
000002A8G	EF	00000098G	EF	D0	00C7D	111\$:	MOVL	IDATA+152, QTAB+680	: 0832
				00V	11	00C88	BRB	113\$	
00000098G	EF			01	D0	00C8A	MOVL	#1, IDATA+152	: 0838
00000000G	EF	00000000G	EF	7D	00C91	113\$:	MOVQ	NULL_STRING, DEFAULT_FILENAME_DESC	: 0840
F8	AD	010E0004	8F	D0	00C9C		MOVL	#17694724, -8(FP)	: 0841
F4	AD	FFFFFF246	EF	D0	00CA4		MOVL	C.AAY, -12(FP)	
FC	AD		AD	9E	00CAC		MOVAB	-12(FP), -4(FP)	
			AD	9F	00CB1		PUSHAB	-8(FP)	
		00000000G	EF	9F	00CB4		PUSHAB	DEFAULT_FILENAME_DESC	
00000000G	EF		02	FB	00CBA		CALLS	#2, STRSTRIM	
00000000G	EF	00000000G	EF	7D	00CC1		MOVQ	NULL_STRING, NL_DEV_DESC	: 0843
F8	AD	010E0003	8F	D0	00CCC		MOVL	#17694723, -8(FP)	: 0844
F4	AD	FFFFFF21A	EF	D0	00CD4		MOVL	C.AAZ, -12(FP)	
FC	AD		AD	9E	00CDC		MOVAB	-12(FP), -4(FP)	
			AD	9F	00CE1		PUSHAB	-8(FP)	
		00000000G	EF	9F	00CE4		PUSHAB	NL_DEV_DESC	
00000000G	EF		02	FB	00CEA		CALLS	#2, STRSTRIM	
00000000G	EF		01	90	00CF1		MOVAB	#1, EDITING	: 0849
				04	00CF8		RET		: 0851

: Routine Size: 3321 bytes. Routine Base: \$CODE + 00206

				0000	00000	INPUT_FDL_FILE:		: 0896
				0000	00000	.WORD	^M<>	
	5E		08	C2	00002	SUBL2	#8, SP	
		F8	AD	D4	00005	CLRL	-8(FP)	
	6D	00000000G	EF	9E	00008	MOVAB	PASSHANDLER, (FP)	
F8	AD	00000000G	EF	9E	0000F	MOVAB	RMS_INPUT_COND_HANDLER, FP-8	: 0903
00V00000000G	EF		00	E0	00017	BBS	#0, AUTO_TONE, 2\$	: 0908
		00000000G	EF	9F	0001F	PUSHAB	SHIFT	: 0910
			04	DD	00025	PUSHL	#4	
		00000000G	EF	9F	00027	PUSHAB	PASS\$V_OUTPUT	
00000000G	EF		03	FB	0002D	CALLS	#3, PASS\$WRITE_STRING	
			01	DD	00034	PUSHL	#1	
	7E	00000000G	EF	9A	00036	MOVZBL	TAB, -(SP)	

EDF  
V04-000

Generated Code

G 13  
16-Sep-1984 01:22:54  
5-Sep-1984 13:37:22

VAX-11 Pascal V2.4-277  
DISK\$VMSMASTER:[EDF.SRC]EDFMAIN.PAS;1 (17) Page 42

00000000G EF  
00000000G EF

01  
01

04  
00

00000000G EF  
00000000G EF  
00V00000000G EF

00000000G

00000000G EF

7E 00000000G

00000000G

00000000G EF

7E 00000000G

00000000G

00000000G EF

FFFFF103

00000000G EF

00000000G

00000000G EF

00000000G

00000000G EF

00000000G

00000000G EF

00000000G

; Routine Size: 307 bytes, Routine Base: \$CODE + 00EFF

0000 00000  
5E 08 C2 00002  
F8 AD D4 00005  
03 00000000G EF 9E 00008  
00 E0 0000F  
0000V 31 00017  
03 00000000G EF 00 E0 0001A  
0000V 31 00022  
F8 AD 00000000G EF 9E 00025  
00V00000000G EF 00 E0 0002D  
00000000G EF 9F 00035  
04 DD 0003B

2\$:

5\$:

PUSHAB PASSFV OUTPUT  
CALLS #3,PASSWRITE\_CHAR  
PUSHL #1  
MOVZBL TAB, -(SP)  
PUSHAB PASSFV OUTPUT  
CALLS #3,PASSWRITE\_CHAR  
PUSHAB C.ABA  
PUSHL #25  
PUSHAB PASSFV OUTPUT  
CALLS #3,PASSWRITE\_STRING  
PUSHAB PASSFV OUTPUT  
CALLS #1,PASSWRITELN2  
CLRB ANALYSIS\_ONLY  
INSV #1,#4,#1,FLAGS  
INSV #1,#0,#1,FLAGS  
PUSHAB FLAGS  
PUSHAB RAB\_DUMMY  
PUSHAB FAB\_DUMMY  
PUSHAB INPUT\_FILENAME\_DESC  
CALLS #4,FDC\$PARSE  
MOVL R0, ISTATUS  
BLBC ISTATUS, 5\$  
BBS #0, AUTO\_TUNE, 5\$  
PUSHAB SHIFT  
PUSHL #4  
PUSHAB PASSFV OUTPUT  
CALLS #3,PASSWRITE\_STRING  
PUSHL #1  
MOVZBL TAB, -(SP)  
PUSHAB PASSFV OUTPUT  
CALLS #3,PASSWRITE\_CHAR  
PUSHL #1  
MOVZBL TAB, -(SP)  
PUSHAB PASSFV OUTPUT  
CALLS #3,PASSWRITE\_CHAR  
PUSHAB C.ABB  
PUSHL #25  
PUSHAB PASSFV OUTPUT  
CALLS #3,PASSWRITE\_STRING  
PUSHAB PASSFV OUTPUT  
CALLS #1,PASSWRITELN2  
RET

: 0915  
: 0921  
: 0922  
: 0928

: 0935  
: 0941

: 0943

.ENTRY INPUT\_ANALYSIS\_FILE, ^M<> : 0988  
SUBL2 #8, SP  
CLRL -8(FP)  
MOVAB PASSHANDLER, (FP)  
BBS #0, EDITING, .+3 : 0995  
BRW 8\$  
BBS #0, ANALYSIS\_SPECIFIED, .+3  
BRW 8\$  
MOVAB RMS\_INPUT\_COND\_HANDLER, FP-8 : 1002  
BBS #0, AUTO\_TONE, 4\$ : 1007  
PUSHAB SHIFT : 1009  
PUSHL #4

ED  
V0  
00  
00  
00  
00  
00  
00  
00  
00

ED  
V0  
00  
00  
00  
00  
00  
00  
00  
00



EDF  
V04-000

H 13  
16-Sep-1984 01:22:54  
5-Sep-1984 13:37:22

VAX-11 Pascal V2.4-277  
DISK\$VMSMASTER:[EDF.SRC]EDFMAIN.PAS;1 (17) Page 43

Generated Code

00000000G	EF	00000000G	EF	9F	0003D	PUSHAB	PASSFV OUTPUT		
00000000G	EF	FFFFF0B2	EF	FB	00043	CALLS	#3,PASSWRITE_STRING		
			15	DD	00050	PUSHAB	C,ABC		
		00000000G	EF	9F	00052	PUSHL	#21		
00000000G	EF	00000000G	03	FB	00058	PUSHAB	PASSFV OUTPUT		
00000000G	EF	00000000G	EF	9F	0005F	CALLS	#3,PASSWRITE_STRING		
00000000G	EF		01	FB	00065	PUSHAB	PASSFV OUTPUT		
00000000G	EF		01	90	0006C	CALLS	#1,PASSWriteln2		
00000000G	EF		00	FB	00073	MOVAB	#1,ANALYSIS_ONLY		: 1014
00000000G	EF		00	FB	0007A	CALLS	#0,POINT-AT-ANALYSIS		: 1015
00000000G	EF		01	F0	00081	CALLS	#0,NEW_IDENT_LINE		: 1020
00000000G	EF		01	F0	0008A	INSV	#1,#4,#1,FLAGS		: 1026
			01	F0	0008A	INSV	#1,#0,#1,FLAGS		: 1027
		00000000G	EF	9F	00093	PUSHAB	FLAGS		: 1033
		00000000G	EF	9F	00099	PUSHAB	RAB_DUMMY		
		00000000G	EF	9F	0009F	PUSHAB	RAB_DUMMY		
		00000000G	EF	9F	000A5	PUSHAB	ANALYSIS_FILENAME_DESC		
00000000G	EF		04	FB	000AB	CALLS	#4,FDL\$PARSE		
00000000G	EF		50	DD	000B2	MOVL	R0,ISTATUS		
		00000000G	EF	94	000B9	CLRB	ANALYSIS_ONLY		: 1040
00000000G	EF		00	FB	000BF	CALLS	#0,POINT-AT-DEFINITION		: 1041
00V00000000G	EF	00V00000000G	EF	E9	000C6	BLBC	ISTATUS,7\$		: 1043
00V00000000G	EF		00	E0	000CD	BBS	#0,AUTO_TUNE,7\$		
		00000000G	EF	9F	000D5	PUSHAB	SHIFT		: 1049
			04	DD	000DB	PUSHL	#4		
00000000G	EF	00000000G	EF	9F	000DD	PUSHAB	PASSFV OUTPUT		
		FFFFF02A	EF	FB	000E3	CALLS	#3,PASSWRITE_STRING		
			17	DD	000F0	PUSHAB	C,ABD		
		00000000G	EF	9F	000F2	PUSHL	#23		
00000000G	EF		03	FB	000F8	PUSHAB	PASSFV OUTPUT		
		00000000G	EF	9F	000FF	CALLS	#3,PASSWRITE_STRING		
			02	DD	00105	PUSHAB	CRLF		
		00000000G	EF	9F	00107	PUSHL	#2		
00000000G	EF		03	FB	0010D	PUSHAB	PASSFV OUTPUT		
		00000000G	EF	9F	00114	CALLS	#3,PASSWRITE_STRING		
00000000G	EF		01	FB	0011A	PUSHAB	PASSFV OUTPUT		
					00121	CALLS	#1,PASSWriteln2		
			04	00121	7\$: 8\$:	RET			: 1053

; Routine Size: 290 bytes, Routine Base: \$CODE + 01032

				0000	00000	SETUP_CONTINUE:		: 1098	
				0000	00000	.WORD	*M<>		
	5E		08	C2	00002	SUBL2	#8,SP		
		F8	AD	D4	00005	CLRL	-8(FP)		
	3D	00000000G	EF	9E	00008	MOVAB	PASSHANDLER,(FP)		
	F8	AD	EF	9E	0000F	MOVAB	CTRLZ_COND_HANDLER,FP-8		: 1105
		00000002	8F	DF	00017	PUSHAL	#2		: 1107
00000000G	EF		01	FB	0001D	CALLS	#1,CLEAR		: 1109
				04	00024	RET			

; Routine Size: 37 bytes, Routine Base: \$CODE + 01154

				0000	00000	DISPATCH_FUNCTION:		: 1163	
				0000	00000	.WORD	*M<>		
	5E		08	C2	00002	SUBL2	#8,SP		



		F8	AD	D4	00005	CLRL	-8(FP)		
		6D	EF	9E	00008	MOVAB	PASSHANDLER,(FP)		
	F8	AD	EF	9E	0000F	MOVAB	CTRLZ_COND_HANDLER,FP-8	:	1170
00V	00000000G	EF	00	E0	00017	BBS	#0,AUTO_TUNE,3\$	:	1172
			AF	9F	0001F	PUSHAB	3\$	:	1174
		00V	19	DD	00022	PUSHL	#25		
		00000000G	EF	9F	00024	PUSHAB	FDL_DEST		
00000000G	EF		03	FB	0002A	CALLS	#3,PASS\$CLOSE2		
00000000G	EF		00	FB	00031	CALLS	#0,POINT_AT_DEFINITION	:	1176
00000000G	EF		01	90	00038	MOVB	#1,DEST_IS_TERMINAL	:	1177
		00000000G	EF	94	0003F	CLRB	OPTIMIZING	:	1178
		00000000G	EF	94	00045	CLRB	VISIBLE_QUESTION	:	1179
		00000000G	EF	94	0004B	CLRB	TEMP_FUCL_PROMPT	:	1180
00000000G	EF	00000000G	EF	90	00051	MOVB	AUTO_TUNE_TAKE_DEFAULTS	:	1181
	07	00000008G	EF	D1	0005C	CMPL	IDATA+8,#7	:	1186
			03	13	00063	BEQL	+3		
		0000V	31	00065	BRW	17\$			
		00000029	8F	DF	00068	PUSHAL	#41	:	1193
08	00000000G	EF	01	FB	0006E	CALLS	#1,QUERY		
	00	000000A4G	EF	CF	00075	CASEL	IDATA+164,#0,#8	:	1195
			0000V		0007D	.DISPL	6\$		
			0000V		0007F	.DISPL	7\$		
			0000V		00081	.DISPL	14\$		
			0000V		00083	.DISPL	8\$		
			0000V		00085	.DISPL	9\$		
			0000V		00087	.DISPL	10\$		
			0000V		00089	.DISPL	11\$		
			0000V		0008B	.DISPL	12\$		
			0000V		0008D	.DISPL	13\$		
		00V	11	0008F	BRB	15\$			
00000000G	EF		00	FB	00091	CALLS	#0,ADD_FDL_LINE	:	1197
		0000V	31	00098	BRW	20\$			
00000000G	EF		00	FB	0009B	CALLS	#0,DELETE_FDL_LINE	:	1198
		00V	11	000A2	BRB	20\$			
00000000G	EF		00	FB	000A4	CALLS	#0,HELP_PROC	:	1199
		00V	11	000AB	BRB	20\$			
00000000G	EF		C0	FB	000AD	CALLS	#0,INVOKE_SCRIPT	:	1200
		00V	11	000B4	BRB	20\$			
00000000G	EF		00	FB	000B6	CALLS	#0,MODIFY_FDL_LINE	:	1201
		00V	11	000BD	BRB	20\$			
		00000000G	EF	94	000BF	CLRB	EDITING	:	1202
			00V	11	000C5	BRB	20\$		
00000000G	EF		00	FB	000C7	CALLS	#0,SET_PROC	:	1203
		00V	11	000CE	BRB	20\$			
00000000G	EF		00	FB	000D0	CALLS	#0,VIEW_DEF	:	1204
		00V	11	000D7	BRB	20\$			
		00000000G	EF	94	000D9	CLRB	EDITING	:	1213
00000000G	EF		00	FB	000DF	CALLS	#0,CREATE_NEW_FDL	:	1214
		00V	11	000E6	BRB	20\$			
		00V	11	000E8	BRB	20\$			
00000108G	EF	00000008G	EF	D0	000EA	MOVL	IDATA+8,IDATA+264	:	1233
			50	94	000F5	CLRB	R0	:	1235
00000100	8F	00000108G	EF	D1	000F7	CMPL	IDATA+264,#256		
		00V	1E	00102	BGEQU	19\$			
00VFFFFEEDB	EF	00000108G	EF	E1	00104	BBC	IDATA+264,C.ABE,19\$		
			50	96	00110	INCB	R0		
00000000G	EF		50	90	00112	MOVB	R0,ISAM_ORG		

**.END**

[illegible]

EDF  
V04-000

Pascal Compilation Statistics

K 13  
16-Sep-1984 01:22:54  
5-Sep-1984 13:37:22

VAX-11 Pascal V2.4-277  
DISK\$VMSMASTER:[EDF.SRC]EDFMAIN.PAS;1 (17) Page 46

COMMAND QUALIFIERS

PASCAL/MACHINE/NODEBUG/NOCHECK/LIS=LIS\$:EDFMAIN/OBJ=OBJ\$:EDFMAIN MSRC\$:EDFMAIN

/CHECK=(NOBOUNDS, NOCASE\_SELECTORS, NOOVERFLOW, NOPOINTERS, NOSUBRANGE)

/DEBUG=(NOSYMBOLS, NOTRACEBACK)

/NOENVIRONMENT

/LIST= \$255\$DUA28:[EDF.LIS]EDFMAIN.LIS;1

/OBJECT= \$255\$DUA28:[EDF.OBJ]EDFMAIN.OBJ;1

/NOCROSS\_REFERENCE /ERROR\_LIMIT=30 /NOG\_FLOATING /MACHINE\_CODE /NOOLD\_VERSION /OPTIMIZE /NOSTANDARD /WARNINGS

COMPILER INTERNAL TIMING

Phase	Faults	CPU Time	Elapsed Time
Initialization	77	00:00.4	00:03.9
Source Analysis	1155	00:22.1	05:09.1
Source Listing	51	00:01.8	00:04.0
Tree Construction	253	00:01.1	00:02.8
Flow Analysis	13	00:00.3	00:00.8
Profit Analysis	51	00:00.5	00:01.7
Context Analysis	255	00:04.6	00:10.1
Name Packing	10	00:00.2	00:00.4
Code Selection	75	00:01.0	00:02.3
Final	140	00:03.9	00:10.5
TOTAL	2083	00:35.8	05:45.8

COMPILATION STATISTICS

CPU Time: 00:35.8 (2198 Lines/Minute)  
Elapsed Time: 05:45.8  
Page Faults: 2083  
Compilation Complete



0127

AH-BT13A-SE  
 VAX/VMS V4.0

**DIGITAL EQUIPMENT CORPORATION**  
**CONFIDENTIAL AND PROPRIETARY**